The Dissertation Committee for Byeongwook Choi
certifies that this is the approved version of the following dissertation:

Numerical Methods for the Valuation of American Options under Jump-Diffusion Processes

Committee:

Patrick Jaillet, Supervisor

Leon Lasdon

Stathis Tompaidis

Hong Yan

Thaleia Zariphopoulou

Numerical Methods for the Valuation of American Options under Jump-Diffusion Processes

by

Byeongwook Choi, B.S., M.B.A.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2002

Dedicated to my wife, Eunjoo Lee.

Acknowledgments

I would like to thank my supervisor, Dr. Patrick Jaillet, for his guidance, insight, and invaluable assistance on this dissertation. Also he offered me invaluable advice on my academic work since I enrolled in the Ph.D. program of the Management Science and Information Systems Department. I am further indebted to my other committee members, Dr. Leon Lasdon, Dr. Stathis Tompaidis, Dr. Hong Yan, and Dr. Thaleia Zariphopoulou for their time, advice, and warm encouragement. I also greatly acknowledge Dr. John Mote for his generous financial support provided to me with teaching assistantship. Thanks are also due to Seowon Jin, who helped me read French articles and gave me many useful comments on my mathematical proofs. My colleagues and friends in the University and Austin also deserve a hearty thanks for all the years of being together with me. Lastly, I should express my gratitude to my parents for their patience and encouragement during my studying in the U.S., and my wife, Eunjoo Lee, for her love.

Byeongwook Choi

Numerical Methods for the Valuation of American Options under Jump-Diffusion Processes

Publication No. ____________

Byeongwook Choi, Ph.D.
The University of Texas at Austin, 2002

Supervisor: Patrick Jaillet

The purpose of this dissertation is studying the numerical valuation of American and European option prices under jump-diffusion processes. Due to the jump part, the market is incomplete and so it is impossible to construct a hedging portfolio with stocks and riskless assets. Contrary to the case of a complete market in which only one equivalent martingale measure exists, there are infinite numbers of equivalent martingale measures in an incomplete market. Our research here is focusing on risk minimizing strategy and its associated minimal martingale measure under the jump-diffusion processes.

Based on this risk minimizing hedging strategy, we characterize the dynamics of a risky asset and derive the valuation formula for an option price. Under the minimal martingale measure, we obtain an analytical formula for a European option price. The main contribution of this dissertation is to extend Kim (1990)'s early exercise premium representation based on a decomposition

method in order to calculate an American option price under jump-diffusion processes as a summation of a European option price and early exercise premiums.

We derive the early exercise premium representation under jump-diffusion processes with various distributions of jump size - lognormal, jump-to-ruin, bivariate and double exponential distribution. In calculating an optimal boundary, we modify and extend numerical methods previously used in the pure diffusion processes - Kim's integral equation method, and Ju's approximation scheme by multipiece exponential functions. Also we apply Richardson extrapolation scheme and modify MacMillan-Zhang's analytical method to calculate American option prices in a faster way.

We implement two previous procedures: a binomial lattice method of Amin (1993) and a semi-implicit finite difference method of Zhang (1997) and compare them with our extended integral equation method. The numerical performance of the extended integral equation method is found to be superior to the previous methods in that the former shows a smaller relative root-mean-square error, possesses a lower degree of algorithmic complexity and converges faster than the two previous methods.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

An option is a derivative security whose value depends on the values of other, more basic underlying assets such as stock, bond, stock index, commodity, and currency. A call option (a put option, respectively) gives its holder the right to buy (to sell, respectively) the option's underlying asset at some future date for a prespecified price. The act of making this transaction is referred to as exercising the option. The holder of an option, however, is not obligated to exercise the right, in that case; the option expires worthless.

Most of the widely traded options in exchanges and over-the-counter (OTC) markets are of American style: such an option allows the holder to have the right to exercise the option at any instant before maturity date. Otherwise it is known as an option of European style, allowing the holder to exercise the option only at its maturity date. The prespecified price is called the strike or exercise price. In order to purchase the option contract, an investor needs to pay an option's price (or premium) to a counter party at the initial date when the contract is entered into. Throughout most of this dissertation we will be discussing valuation issues for an option written on a single share of common stock, especially for an American style option.

The value of an American option is greater than or equal to that of a European counterpart since there are more exercising opportunities in an American option than in European one. The value of an American call option on a non-dividend paying stock is equivalent to that of a European counterpart since it is known that exercising the American call option before the maturity date is never optimal. In general, the value of an American option might be decomposed in two terms: a European counterpart and an early exercise premium.

For the holder of an American option, the exercise time of an option is a crucial decision making problem. Intuitively, one might expect that the holder of an American option chooses his or her exercise policy in such a way that the expected payoff from the option is maximized. For example, in exercising an American call option written on a dividend paying stock, one should consider not only dividends from the stock that will be received until the maturity date, but also interests which otherwise can be earned on the exercise price. The optimal exercise of the American option can be characterized by optimal stopping boundary or optimal exercise boundary. The optimal exercise boundary is a set of critical stock prices under which for an American put (or over which for an American call) it is optimal to exercise the option immediately. Since it is straightforward to compute an American option price, once the optimal exercise boundary is known, obtaining the critical stock prices efficiently is a possible strategy for pricing American options and it is one of the objectives of this paper.

One of the assumptions previously made in earlier works on the valuation of an option is that the underlying stock price follows a geometric Brownian motion (or diffusion process) through time which produces a log-normal distribution for the stock price between any two points in time. The diffusion processes provide a nice framework to analyze lots of financial derivatives mathematically and simplify the analysis with a relatively-less complicated Itó stochastic calculus. The option pricing models of Black-Scholes (1973) and Merton (1973) are derived in this framework. However, when the true return distribution of the underlying asset shows asymmetric leptokurtic features such as a high peak or heavier tails, the option price may be mispriced. Another drawback is the so-called volatility smile: the implied volatility of an option as a function of its strike price resembles smile curve. However it should be constant in a framework of Black-Scholes (1973) model. To overcome these drawbacks, we will look at jump-diffusion processes as an alternative model introduced by Merton (1976).

Under the jump-diffusion processes, the underlying asset is allowed to have jumps which imply that there is a positive probability of a stock-price change of some magnitude no matter how small the time interval between successive observations. The security market is no longer complete in a sense that we cannot construct a dynamic portfolio to replicate the value of an option exactly. In other words, not every derivative's price can be spanned by the existing assets such as stocks and bonds. Contrary to a complete market in which only one equivalent martingale measure (or a risk-neutral measure) exists, in an incomplete market there are infinite numbers of equivalent martingale measures, and the risk exposure cannot always be eliminated completely by means of a judicious trading strategy.

Using the risk minimizing trading strategy proposed by Föllmer, Schweizer and Sondermann (1986, 1990, and 1991) and under the minimal martingale measure driven by the strategy, we derive a pricing formula of a European option and an American option under jump-diffusion processes. In a simple case where we assume that the actual probability measure is a martingale measure (i.e., the actual world is risk-neutral), the option prices obtained from the risk-minimizing strategy coincide with those derived by Merton (1976). We find however that the ways of "balancing" market prices of two risk sources (diffusion and jump parts) between the two models are totally different. In other words, the Radon-Nikodym derivative process (also called a pricing kernel or a stochastic discount factor) of the simple model is different from that of Merton's model, although the corresponding option pricing formulas are the same.

In the more complicated case (i.e., the actual world is not risk-neutral), the value of an option can be defined only when the Radon-Nikodym derivative process is positive, and it is achievable when the relative jump sizes of an underlying asset price are bounded above by some levels. We discuss several issues arising from the restriction in applying the risk minimizing strategy to the valuation of option prices.

In addition to the contribution to methodology discussed in the previous paragraph, the main contribution of this dissertation is on the development and analysis of alternative numerical procedures to price American options under the jump-diffusion model. We decompose the American option prices into two parts under jump-diffusion processes: an equivalent European option price and an early exercise premium. This decomposition is based on the methodology of Kim (1990) who introduced the idea for pure diffusion processes (see also Jacka (1991) and Carr, Jarrow and Myneni (1992)). The decomposition gives nice intuition for the price of an American option and simplify the valuation problem into solving integral equations where critical stock prices are computed recursively. We obtain the valuation formulas of American options with various distributions of jump size such as jump-to-ruin, bivariate, and double exponential distribution.

We implement our new proposed method and compare it with two existing methods: a binomial lattice with jumps method introduced by Amin (1993) and a semi-implicit finite difference method, which is a variational inequality version for the jump-diffusion cases developed by Zhang (1997). The common characteristics of the above numerical approaches to solving the American option prices is that they approximate the continuous model by discretizing the relevant equations in both the time space and the space for stock prices. For example, binomial lattice with jumps method divides the time to maturity into N equally spaced intervals and for each instance it computes the option price with some restrictions such as no-arbitrage opportunities and stock movements. Therefore, as the number of times discretized is larger, the value will get closer to the true value. The integral equation method uses a trapezoidal rule in order to approximate the integral. The finite difference method discretizes the stock price as well as the time to maturity, and replaces the partial differential equations with finite difference equations by approximations based on Taylor series expansions of functions.

The third part of our research uses several numerical methods previously applied to the diffusion processes - Richardson extrapolation scheme of Geske and Johnson (1984), and the approximation method by multi-piece exponential functions of Ju (1998) - to approximate the critical stock prices in a more efficient way. We implement and evaluate the approximation methods to see whether they are still effective for the model in the presence of jumps.

In order to compare the efficiency of the numerical methods, we focus on three aspects of each algorithm: (1) complexity, (2) speed of convergence, and (3) accuracy. We analyze the algorithms by evaluating the option prices and CPU running times of the computer programs implemented by ANSI C language under a UNIX environment.

The outline of this dissertation is as follows. In Chapter 2, we review previous works on the valuation of American options, and on the valuation issues in incomplete markets, including the case for which the underlying stock price follows a jump-diffusion process. In Chapter 3 we present the necessary frameworks, make assumptions, define the notations, introduce the minimal martingale measure, and discuss several related issues on option pricing problems in the presence of jumps under the minimal martingale measure. This is followed by a derivation of the pricing formula of a European option under the minimal martingale measure. We begin in Chapter 4 with a derivation of the early exercise premium representation for American options under jump-diffusion processes with various distributions of jump size. In Chapter 5, we implement our extended integral equation method as well as existing numerical procedures and compare the option prices obtained from the various methods. In this chapter, we also analyze the numerical efficiency of the three models and discuss the pricing of American option in higher dimensions. Finally, Chapter 6 concludes this dissertation.

Chapter 2

Literature Review

2.1 Valuation of American options

While a European option price under geometric Brownian motion model is easily computed analytically by using the famous Black-Scholes (1973) formula, an American style option depends on more complicated numerical approaches to compute its price. In general, the valuation of an option can be formulated as a Partial Differential Equations (PDE) under the assumption that the underlying asset price is driven by a diffusion process in a frictionless market under no-arbitrage opportunity. Unfortunately for the American options, the PDE is a free boundary problem (McKean (1965) and Merton (1973)), whose analytic solution, if any, has not yet been obtained. Merton (1973) proves that the value of the American call options on a non-dividend paying stock is equal to the equivalent European options by showing that it is never optimal for the buyer of the options to exercise it before maturity date.

There are many approximate approaches to the valuation of American options. These methods can be classified into three groups. The first group includes numerical techniques such as the binomial tree model of Cox, Ross and Rubinstein (1979), Monte Carlo simulation method of Boyle (1977), Broadie, and Glasserman (1996), and the finite difference methods of Brennan and Schwartz (1977). Inequality model of Jaillet, Lamberton and Lapeyre (1990) is an approach to change the free boundary PDEs into variational inequalities with some boundary conditions for computational purposes. In applying the finite difference approximations to American option pricing problems, we can classify the method into implicit finite difference scheme, explicit finite difference scheme, and Crank-Nicholson scheme according to the direction to which the first derivative is approximated. Dempster and Hutton (1999) solve the finite difference equations by converting it as a linear programming formulation. These are traditional methods and widely used in pricing American options as well as European options since it is simple to implement and the numerical solution converges to the true value. However, they are in general very time consuming.

The second approach is to approximate the analytical formula of the American option price. This group includes Johnson (1983), MacMillan (1986), and Barone-Adesi and Whaley (1987). Their methods yield the price easily and quickly, but are less accurate. Even worse, there is no way to make the solution convergent to the exact solution.

The third group uses early exercise premium representation of American option price derived by Kim (1990), Jacka (1991), and Carr, Jarrow, and Myneni (1992). According to this representation, the value of the American put option is decomposed into that of an equivalent European put option and the early exercise premium. However the major difficulty in evaluating the decomposition formula arises from the fact that it is very laborious to calculate the critical stock price, which is defined as a stock price, below (above, respectively) which it is optimal to exercise the American put (call, respectively). In some papers, the critical stock is called an optimal exercise boundary. Once the critical stock price is obtained however, it is straightforward to calculate the American option price and relevant hedge parameters.

2.2 Valuation in incomplete markets

If the underlying asset follows a jump-diffusion process, we cannot construct a dynamic portfolio to replicate the payoff of an option perfectly (i.e., perfect replication), and this security market is called incomplete. In an incomplete market, a contingent claim is not necessarily a stochastic integral of a stock price process and one can only hope to find the best strategy based on one's "reasonable" criterion. We introduce here only some of the related works in order to keep the scope of this survey manageable.

One of the approaches to handle a valuation problem in incomplete market is based on the local risk minimizing strategy as discussed in several papers by Föllmer, Schweizer, and Sondermann (Föllmer and Sondermann (1986), Föllmer and Schweizer (1990), and Schweizer (1991)). The local risk minimizing strategy allows cash-inflows or outflows and looks for an admissible trading strategy that minimizes the magnitude of cash-flows. An equivalent martingale measure derived from this strategy is called the minimal martingale measure. Colwell and Elliott (1993) derive and explore the characteristics of the minimal martingale measure for the jump-diffusion model by applying the local risk minimizing strategy to a valuation problem in the presence of jumps.

Another available trading strategy is the variance minimizing hedging strategy used by Duffie and Jackson (1990), Duffie and Richarson (1991), Schäl (1994), Schweizer (1992, 1995, 1996), and Bertsimas, Kogan and Lo (2001). This strategy searches for an optimal self-financing strategy that minimizes the expected quadratic terminal risk with or without a prespecified initial investment. Bertsimas, Kogan, and Lo (2001), applying stochastic dynamic programming to the minimization of a mean variance function under Markov state-dynamics, derive recursive expression for the optimal replicating strategy. They show that the replicating cost that minimizes the mean variance function under an equivalent martingale measure corresponds to the equilibrium price of the option.

El Karoui and Quenez (1995) assert that there is a price range for the actual market price of an option in an incomplete market and study its maximum and minimum price using stochastic control methods. One of their findings is that the maximum price is the selling price defined as the smallest price that allows the seller to hedge completely by a trading portfolio. A similar result is obtained for the minimum price.

Davis, Panas, and Zariphopoulou (1993), Davis and Zariphopoulou (1995), and Constantinides and Zariphopoulou (2001) suggest an option price as the maximum price at which a utility-maximizing investor would include the option in his or her portfolio. They characterize the fair prices for European and American options based on utility maximization in the presence of transaction costs. They transform this problem into that similar to Merton's original problem (1969, 1971) of optimal consumption and portfolio choice for a single investor in an intertemporal economy. The investor's goal is to maximize his or her expected utility from terminal wealth and/or the expected utility of intermediate consumption and the goal is characterized by a value function. They then derive Hamilton-Jacobi-Bellman (HJB) equations for an option pricing problem under a suitably chosen utility function. As pointed out by Pham (1998), however, there is not in general a smooth solution of the HJB equation especially when the diffusion coefficients is degenerate. Therefore one is forced to use a notion of weak solutions such as viscosity solutions which has been first introduced to finance by Zariphopoulou (1999).

2.3 Valuation under jump-diffusion processes

The classical log-normal diffusion model of Black-Scholes (BS) cannot handle the "rare events" which has to do with the discontinuity of the observed price processes. There are several empirical researches supporting the presence of jumps especially in the interest rates, foreign exchange rates, and energy commodity price (see Ahn and Thomas (1988), Jorion (1988), Bates (1996) and Deng (1998)). Also jump model might explain the mispricing with respect to the Black-Scholes model (see Bakshi, Cao and Chen (1997)).

In the presence of jump processes, the derivation of a formula for the price of derivatives is complicated by several factors. First, one must need the extension of the Ito's lemma for the jump-diffusion process (Merton, 1990). This leads to an expression for the derivative price which is a nonlinear function of underlying assets. Second this previous fact makes it impossible to construct a riskless portfolio to replicate the price of derivatives and therefore the corresponding economic model is not complete. For the second reason, some researchers heavily rely on equilibrium argument instead of no-arbitrage arguments, as in Ahn and Thomson (1988), Naik and Lee (1990), and Attari (1999).

There are few papers on the valuation of derivatives in the presence of jump-diffusion processes. Merton (1976) has proposed a model where in addition to a Brownian motion term, the price process of the underlying is allowed to have jumps. With the assumption that the jump component of the stock's return is due to non-systematic risk, and therefore this component earns the risk-free rate of return under the Capital Asset Pricing Model, or CAPM, the prices of European options can be easily obtained. When the jump size is assumed to be a log-normal distribution, Merton (1976) presents closed form solution for the European option prices.

Naik and Lee (1990) and Ahn (1992) use a general equilibrium framework to price options on the market portfolio with discontinuous returns by embedding the option-pricing problem in a representative agent economy of the Lucas (1978) type, pointing out that Merton's assumption that the jumps in security prices are uncorrelated with return on the market portfolio is violated if the security under consideration is the market portfolio. They assert that if the trading is allowed only in the underlying asset and a riskless bond, pricing of options on that asset by no-arbitrage arguments is not possible.

Amin (1993) provides a method to price the European and American options using discrete-time models proposed by Cox, Ross, and Rubinstein (1979). This method expresses the price of a derivative as a recursive function of its price at the previous time step and obtains the price with a back-ward dynamic programming. Amin also discusses early exercise behavior by looking at critical stock prices for the two types of jumps: (1) bankruptcy-inducing jumps, and (2) log-normal jumps. With bankruptcy-inducing jumps, he finds that early exercise is postponed for both puts and calls relative to the Black-Scholes model for all maturities. With log-normal jumps, however, he observes that unlike the early exercise decisions in the Black-Scholes model, the possibility of jumps causes early exercise to be postponed when the time to maturity is small and to be accelerated when it is long for American options.

Zhang (1994, 1997) applies the variational inequality techniques developed in Jaillet, Lamberton, and Lapeyre (1990) for the American option pricing problem to the same problem but with jump-diffusion processes. Like Amin (1993), her research presents numerical examples of American option prices under jump-diffusions. Pham (1997) and Gukhal (2001) obtain a decomposition of the American put option price as the sum of its corresponding European put price and the early exercise premium. Compared with the early exercise representations when jumps are not allowed (Kim (1990), Jacka (1991), and Carr, Jarrow and Myneni (1992)), this representation has an extra complex term due to the jumps, and involves American option prices recursively, which makes it impossible to implement numerically.

2.4 Valuation of options on multiple underlying assets

In this section we review several models developed for American options on multiple underlying assets under the pure diffusion process. There are many financial products whose value depends on two or more underlying assets. The typical examples are options on the average of several underlying assets, options on the maximum of several underlying assets, options on the difference of two assets, quanto options, and Bermudan options on swaps (swaptions).

Handling of multiple state variables (i.e., the price processes of several underlying assets) becomes much more difficult due to the early exercise opportunities of American derivatives. There are several approaches which extend those for standard American options.

- Multinomial lattice method extended from the binomial model of Cox, Ross and Rubinstein (Boyle, Evnine and Gibbs, 1989).
- Integral equation model extending the approach that decomposes the American option price into the European counter part and early exercise premium (Broadie and Detemple (1997), Villeneuve (1999)).
- Alternating direction implicit (ADI) method extending the finite difference method to higher dimensions (Mitchell and Griffiths (2001), and Villeneuve and Zanette (2002)).

- Simulation method using primal-dual property by Haugh and Kogan (2001), and Anderson and Broadie (2001).

Multinomial lattice model:

Generalization of the binomial model of Cox, Ross, and Rubinstein to the multi-dimensional model was suggested by Boyle (1988), Boyle, Evnine, and Gibbs (1989), and He (1990). Especially, Boyle, Evnine, and Gibbs (1989) developed a numerical method for options involving multiple underlying assets by successfully obtaining a closed form solution of the jump sizes and jump probabilities in a multiple lattice framework so that the characteristic function of the discrete distribution converges to that of the continuous distribution. Their method first fix the probability of an up-jump and then determine jump sizes to ensure convergence. Numerical examples for European option prices on the maximum, minimum, geometric average and arithmetic average of three assets are presented. They comment that the prices of the American counter part can be readily handled with the same approach. However as Anderson and Broadie (2001) point out, the computational effort of the multinomial lattice procedure grows exponentially with the number of state variables, and so the method is impractical for higher dimensional problems.

Integral method:

We are considering the case of American options on multiple underlying assets. The typical option of this type is an American option on the maximum of two assets. The prices of the underlying assets at time t, S_t^1,a nd S_t^2, are given by the stochastic differential equations

$$\begin{aligned} dS_t^1 &= S_t^1(r-\delta^1)dt + S_t^1\sigma_1 dW_t^1 \\ dS_t^2 &= S_t^2(r-\delta^2)dt + S_t^2\sigma_2 dW_t^2 \end{aligned} \tag{2.1}$$

where r is the constant risk-free interest rate, $\delta_i \geq 0$ is the dividend rate, and σ_i is the volatility of the price of asset i, $i = 1, 2$. W_t^1 and W_t^2 are standard Brownian motion under the risk-neutral measure $\mathbb{P}^*$ with a constant correlation ρ. Thus the above price processes are satisfied under the risk-neutral economy.

Let $C_t(S_t^i)$ define the value of American call option at time t on a single asset i (standard option) that matures at time T and has a exercise price K, and let $C_t(S_t^1, S_t^2)$ denote the value of American call option on the maximum of two assets (max-option). The payoff of max-option at any time t before maturity T is defined by $[\max(S_t^1, S_t^2) - K]^+$.

As Broadie and Detemple (1997) show, the optimal exercise boundary is determined by the level of two stock prices S_t^1 and S_t^2. Let $B_1(S_t^2, t)$ and $B_2(S_t^1, t)$ be the optimal boundary on the two-dimensional (S_t^1, S_t^2) plane, respectively. Broadie and Detemple (1997) also show that the value of American max-option can be decomposed into the European max-option price and the the early exercise premium and it is given by

$$C_t(S_t^1, S_t^2) = c_t(S_t^1, S_t^2) + e_1(S_t^1, S_t^2, B_1(\cdot,\cdot)) + e_2(S_t^1, S_t^2, B_2(\cdot,\cdot)), \tag{2.2}$$

where $c_t(\cdot,\cdot)$ is the value of European counterpart, and e_1 and e_2 are the early exercise premiums defined by

$$
\begin{aligned}
e_1(S_t^1, S_t^2, B_1(\cdot,\cdot)) &= \int_t^T e^{-r(v-t)} \mathbb{E}^{\mathbb{P}^*}[(\delta_1 S_v^1 - rK) 1_{\{S_v^1 > B_1(S_v^2, v)\}}]\, dv \\
e_2(S_t^1, S_t^2, B_1(\cdot,\cdot)) &= \int_t^T e^{-r(v-t)} \mathbb{E}^{\mathbb{P}^*}[(\delta_2 S_v^2 - rK) 1_{\{S_v^2 > B_2(S_v^1, v)\}}]\, dv
\end{aligned}
\tag{2.3}
$$

In order to evaluate the above option price, one must obtain the optimal exercise boundary $B_1(S_t^1, S_t^2)$ and $B_2(S_t^1, S_t^2)$, and they are determined by solving the system of recursive integral equations

$$
\begin{aligned}
B_1(S_t^2, t) - K &= c_t(B_1(S_t^2, t), S_t^2) + e_1(B_1(S_t^2, t), S_t^2, B_1(\cdot,\cdot)) \\
&\quad + e_2(B_1(S_t^2, t), S_t^2, B_1(\cdot,\cdot)) \\
B_2(S_t^1, t) - K &= c_t(S_t^1, B_2(S_t^1, t)) + e_1(S_t^1, B_2(S_t^1, t), B_1(\cdot,\cdot)) \\
&\quad + e_2(S_t^1, B_2(S_t^1, t), B_2(\cdot,\cdot))
\end{aligned}
\tag{2.4}
$$

subject to the boundary conditions

$$
\begin{aligned}
\lim_{t\to T} B_1(S_t^2, t) &= \max(\hat{B}_T^1, S_T^2) \\
\lim_{t\to T} B_2(S_t^1, t) &= \max(\hat{B}_T^2, S_T^1)
\end{aligned}
\tag{2.5}
$$

$$
B_1(0, t) = \hat{B}_t^1, \qquad B_2(0, t) = \hat{B}_t^2 \tag{2.6}
$$

Compared with the integral equations for the standard option, the optimal boundary which can be obtained from the solution of the Equations (2.4), (2.5), and (2.6) requires evaluation with respect to the stock price as well as time. Villeneuve (1999) also discusses on exercise regions of American options and characterizes the nonemptiness of the exercise regions.

Alternating Direction Implicit (ADI) method:

The alternating direction implicit (ADI) algorithm is developed by Peaceman and Rachford (1955) for efficiently solving a large-scale system of linear equations arising from the finite differences discretization of elliptic or parabolic equations. Peaceman and Rachford (1955) apply the ADI method to solve the linear complementarity problem (LCP) arising from the discretization of the parabolic variational inequalities related to the problem of American option pricing. The ADI algorithm is based on the LU decomposition for tridiagonal matrices.

Assume the value of American Max-option C has partial derivatives $\partial C/\partial S^i$, $i = 1, 2$, which are uniformly bounded and $\partial C/\partial t$ and $\partial^2 C/\partial S^i \partial S^j$, $i, j = 1, 2$, which are locally bounded on $[0, t) \times \mathbb{R}^+ \times \mathbb{R}^+$. Define the operator $\mathfrak{L}$ on the value function C by

$$\begin{aligned}\mathfrak{L}C =& (r-\delta_1)S^1\frac{\partial C}{\partial S^1} + (r-\delta_2)S^2\frac{\partial C}{\partial S^2}\\ +& \frac{1}{2}\left[\sigma_1^2(S^1)^2\frac{\partial^2 C}{(\partial S^1)^2} + 2\rho\sigma_1\sigma_2\frac{\partial^2 C}{\partial S^1\partial S^2} + \sigma_2^2(S^2)^2\frac{\partial^2 C}{(\partial S^2)^2}\right] - rC.\end{aligned}$$

Then Broadie and Detemple (1997) show that $C_t(S_t^1, S_t^2)$ satisfies the following variational inequalities:

$$\begin{aligned}&C_t \geq [\max(S_t^1, S_t^2) - K]^+, \qquad \frac{\partial C}{\partial t} + \mathfrak{L}C \leq 0\\ &\left(\frac{\partial C}{\partial t} + \mathfrak{L}C\right)\left([\max(S_t^1, S_t^2) - K]^+ - C_t\right) = 0\end{aligned} \tag{2.7}$$

almost everywhere on $[0, T) \times \mathbb{R}^+ \times \mathbb{R}^+$. As pointed out by Boyle, Evnine, and Gibbs (1989), solving partial differential equations involving with more than two-dimensional case is excessively burdensome. Like the multinomial lattice model, the computational effort of the multi-dimensional finite difference method grows exponentially with respect to the number of state variables.

Simulation method:

Boyle (1977) first introduced Monte Carlo simulation for the pricing of European options, and Bossaerts (1989) and Tilley (1993) applied the simulation method to the valuation of American options. Broadie and Glasserman (1997a,b) generate lower and upper bound of the American options using Jensen's inequality. Boyle, Kolkiewicz and Tan (2001)generalize the approach of Broadie and Glasserman (1997b) using low discrepancy sequences to improve the numerical efficiency.

Haugh and Kogan (2001) proposed upper and lower bound using any approximation to the option price. The tightness of bounds depends on the degree to which the initial approximation is close to the true option price. They represent the American option price as a solution of a dual minimization problem. Using this primal-dual result, they simulate the suboptimal exercise strategy implied by the approximate option price and a different stochastic process determined by an appropriate supermatingale, and obtain the lower and upper bound, respectively. Furthermore their method relies on low discrepancy sequences instead of Monte Carlo simulation in estimating the continuation value of the option.

The method of Anderson and Broadie (2001) is very similar to that of Haugh and Kogan (2001). In deriving an upper bound, however the former method involves only straightforward Monte Carlo simulation rather than low discrepancy sequences and uses only the information from the approximation to the optimal exercise strategy instead of approximate option price use by Haugh and Kogan (2001).

Chapter 3

Framework for Valuation in Incomplete Markets

In the first section, we make some assumptions on a market, tradable assets, price processes of the assets, and the associated parameters on a probability space. We begin in Section 2 with an introduction of Merton's model (1976). This is followed by a characterization of the equivalent martingale measures and discussion on the minimal martingale measure proposed by Föllmer, Schweizer, and Sondermann (Föllmer and Sondermann (1986), Föllmer and Schweizer (1990), and Schweizer (1991); hereafter we call the authors FSS). Next, we derive a formula for European option prices in the presence of jumps under the minimal martingale measure. This is new in the literature as we know. We also discuss a simple model in which the actual probability measure is a risk-neutral measure, and compare it with Merton's model.

3.1 Assumptions

We assume throughout this thesis that (1) the capital markets are frictionless, and trading takes place continuously and without transaction costs, (2) there are two tradable assets in the market, a risky asset and a riskless asset, (3) the short-term interest rate is known and constant through time, and (4) the stock price follows a jump-diffusion process through time. The riskless asset B_t at time t is governed by the equations $dB_t = rB_t dt$, and $B_0 = 1$. We assume that the price of risky asset S_t is described by a stochastic differential equation of the form:

$$\frac{dS_t}{S_{t^-}} = \mu dt + \sigma dW_t + d\left(\sum_{j=1}^{N_t} U_j\right) \tag{3.1}$$

To be more rigorous, we consider a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with $\mathbb{F} = (\mathcal{F}_t)_{t\in[0,T]}$, a filtration satisfying the usual conditions on which we define a standard Brownian motion $(W_t)_{t\geq 0}$, a Poisson process $(N_t)_{t\geq 0}$ with jump intensity λ (the average number of arrivals per unit time) and a sequence of $(U_j)_{j\geq 1}$ of independent, identically distributed random variables taking values in $(-1, +\infty)$. We assume that the σ-algebras generated respectively by $(W_t)_{t\geq 0}, (N_t)_{t\geq 0}, (U_j)_{j\geq 1}$ are independent.[1] For simplicity, we take the drift μ and the volatility σ to be a constant and assume that the asset pays no

[1]Merton (1976) points out that to be consistent with the general Efficient Market Hypothesis of Fama (1970) and Samuelson (1965), the dynamics of the unanticipated part of the risky asset should be a martingale. Thus in the papers of Merton (1976), Colwell & Elliott (1993) and Pham (1997), the authors use the compensated jump martingale for jump processes. In other words, they assume that the risky asset price process is given by

$$\frac{dS_t}{S_{t^-}} = (\mu' - \lambda \mathbb{E}U_1)dt + \sigma dW_t + d\left(\sum_{j=1}^{N_t} U_j'\right)$$

However if we regard μ in Equation (3.1) as the net expected rate of return on the risky asset excluding the corresponding rate of jump parts, the two dynamics of stock prices are equivalent.

dividend. Then the dynamics of $(S_t)_{t\geq 0}$ is given by

$$\begin{aligned} S_t &= S_0 \exp\left[(\mu - \frac{\sigma^2}{2})t + \sigma W_t\right] \prod_{j=1}^{N_t}(1+U_j) \quad \text{or,} \\ &= S_0 \exp\left[(\mu - \frac{\sigma^2}{2})t + \sigma W_t + \sum_{j=1}^{N_t} \ln(1+U)_j)\right] \end{aligned} \tag{3.2}$$

where $\prod_{j=1}^{0} = 1$.

We define the discounted stock prices as $\tilde{S}_t = e^{-rt}S_t$. The quadratic variation of $\tilde{S}_t$ is given by:

$$\begin{aligned} \langle \tilde{S}_t \rangle &= \int_0^t \tilde{S}_u^2(\sigma^2 + \lambda \mathbb{E}U_1^2)du \\ &= \sigma_{total}^2 \int_0^t \tilde{S}_u^2 du \end{aligned}$$

Thus we can decompose the total variance of stock price return into the variance of diffusion component and that of jump component:

$$\sigma_{total}^2 = \sigma^2 + \lambda \mathbb{E}U_1^2. \tag{3.3}$$

If we assume that $\ln(1+U_j)$ follows $\mathcal{N}(m, \delta^2)$, and let $k = \mathbb{E}U_j$, then $1+k = \mathbb{E}(1+U_j) = e^{\delta^2/2+m}$, and so:

$$\begin{aligned} \mathbb{E}U_j^2 &= \mathbb{E}(1+U_j)^2 - \mathbb{E}(1+2U_j) \\ &= \mathbb{E}e^{2\ln(1+U_j)} - (1+2k) \\ &= e^{2\delta^2+2m} - (1+2k) \\ &= (k+1)^2 e^{\delta^2} - 2k - 1. \end{aligned} \tag{3.4}$$

Therefore, the total variance can be denoted by:

$$\sigma_{total}^2 = \sigma^2 + \lambda[(k+1)^2 e^{\delta^2} - 2k - 1] \tag{3.5}$$

Now consider a trading strategy $\varphi = (\pi_t^0, \pi_t)$ where π_t^0 and π_t are the amount of riskless assets and stocks holding at time t, respectively. We assume that the trading strategy satifies (i) π_t is $\mathcal{F}_t$-predictable, (ii) π_t^0 is adapted, (iii) $\mathbb{E}[\int_0^T \pi_t^2 d\langle \tilde{S}_t \rangle + (\int_0^T |\pi_t \mu_t| dt)^2] < \infty$, and (iv) the *discounted value process* $V_t(\varphi) \equiv \pi_t \tilde{S}_t + \pi_t^0$ has the right continuous sample paths and $V_t(\varphi) \in L^2(\Omega, \mathbb{P})$ for $0 \leq t \leq T$.

The *discounted gain process* from a trading strategy is defined to be $G_t(\varphi) \equiv \int_0^t \pi_u d\tilde{S}_u$, and the *discounted cost process* is $C_t(\varphi) \equiv V_t(\varphi) - G_t(\varphi)$. A trading strategy φ such that $C_t(\varphi) = C_0(\varphi)$ for all $0 \leq t \leq T$ is called *self-financing.* If, for some contingent claim $X \in L^2(\Omega, \mathbb{P})$, there is a self-financing trading strategy such that $V_T(\varphi) = X$, that is $X = C_0(\varphi) + \int_0^T \pi_u d\tilde{S}_u$ a.s., then φ is a riskless hedge portfolio for X and X is said to be *attainable.* If every contingent claim $X \in L^2(\Omega, \mathbb{P})$ is attainable, the market is called *complete.*

3.2 Merton's approach

Merton (1976) overcomes this valuation problem by assuming that the jump component of the stock's return represents non-systematic risk[2]. According to the CAPM, the expected return on all zero-beta securities must

[2]The risk that can potentially be eliminated by diversification is called unsystematic risk. But there is also some risk that you can't avoid, regardless of how much you diversify. This risk is generally known as a systematic risk (Brealey and Myers, 1991, p.137). The unsystematic risk, also called as an "idiosyncratic risk" in Cochrane (2001), is described to be uncorrelated with the stochastic discount factor and generates no premium. Ingersoll (1987) also mentions that in the CAPM nonsystematic risk is the portion of returns uncorrelated with the return on the market portflio

equal the riskless rate[3], and then he shows that the European option price $f(t) = F(S_t, T - t)$, which is a twice-continuously differentiable function of the stock and time, satisfies

$$0 = \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 F}{\partial S^2} + (r - \lambda \mathbb{E} U_1) S \frac{\partial F}{\partial S} - \frac{\partial F}{\partial \tau} - rF + \lambda \mathbb{E}\{F(S(1+U_1), \tau) - F(S, \tau)\} \tag{3.6}$$

subject to the boundary conditions

$$\begin{aligned} F(0, \tau) &= 0 \\ F(S, 0) &= \max(0, S - K), \end{aligned} \tag{3.7}$$

where K is the exercise price of the option. Note here that Equation (3.6) does not depend on μ. Merton also shows that the solution to Equation (3.6) for the European option price, when the current stock price is S, is given by:

$$F(S, \tau) = \sum_{n=0}^{\infty} \frac{e^{-\lambda\tau}(\lambda\tau)^n}{n!} \, \mathbb{E} f^{BS}\left(S \exp\{-\lambda \mathbb{E} U_1 \tau\} \prod_{j=1}^{n}(1 + U_j), \tau; K, \sigma^2, r\right) \tag{3.8}$$

where $\tau = T - t$ and $f^{BS}(S, \tau; K, \sigma^2, r)$ is the Black-Scholes option pricing formula for the no-jump case. If we assume that the size of the proportional jump has a log-normal distribution, then $\prod_{j=1}^{n}(1 + U_j)$ will have a log-normal distribution with the variance of the logarithm of $\prod_{j=1}^{n}(1 + U_j)$ equal to $n\delta^2$, where δ^2 denotes the variance of $\ln(1 + U_j)$. In this special case, Merton shows that the value of European option is given by[4]

$$F(S, \tau) = \sum_{n=0}^{\infty} \frac{e^{-\lambda'\tau}(\lambda'\tau)^n}{n!} f^{BS}(S, \tau; K, \nu_n, r_n) \tag{3.9}$$

[3] Jarrow and Rosenfeld (1984) point out that jump risk for the stock is not diversifiable and instantaneous CAPM assumed by Merton (1976) will not hold.

[4] See Appendix for proofs.

where $\lambda' = \lambda\mathbb{E}(1+U_j)$. The variable $f^{BS}(S,\tau;K,\nu_n,r_n)$ is the Black-Scholes option price, conditional on knowing that exactly n Poisson jumps will occur during the life of the option when the conditional variance rate (ν_n^2) is $\sigma^2 + \frac{n\delta^2}{\tau}$ and the conditional risk-free rate (r_n) is $r - \lambda\mathbb{E}U_1 + \frac{\log(1+\mathbb{E}U_1)^n}{\tau}$.

In the last section of this Chapter, we show that the above formula for a European option is equivalent to the one obtained when the risk-minimizing trading strategy is used with the restriction that the actual probability measure is considered a martingale measure.

3.3 Minimal martingale measure

In this section, using the risk minimizing trading strategy proposed by FSS and under the minimal martingale measure driven by the strategy, we derive a pricing formula of a European option and discuss some issues on the valuation of an American option under jump-diffusion processes. To avoid any confusion, we let $\mathbb{P}^*$ denote the equivalent martingale measure and $\mathbb{Q}$ the minimal martingale measure respectively. Note that the minimal martingale measure is one of equivalent martingale measures.

We assume here the discounted stock price is a semi-martingale. In this case, a local risk minimizing strategy (associated with *the minimal martingale measure*) is proposed by FSS. They look for an admissible trading strategy which minimizes, at t, the remaining risk

$$R_t(\varphi) \equiv \mathbb{E}^{\mathbb{P}}[(C_T(\varphi) - C_t(\varphi))^2|\mathcal{F}_t] \tag{3.10}$$

where C_t is a cost process at t defined earlier in the previous section.

Suppose the price process $\tilde{S}_t$ is a semi-martingale with the Doob-Meyer decomposition:

$$\tilde{S}_t = \tilde{S}_0 + M + A \tag{3.11}$$

where $M = (M_t)_{0 \le t \le T}$ is a square integrable martingale under $\mathbb{P}$, and $A = (A_t)_{0 \le t \le T}$ is a predictable process with paths of bounded variation such that $A_t = \int_0^t \alpha_s d\langle M \rangle_s$ for some predictable process $\alpha = (\alpha_t)_{0 \le t \le T}$.

FSS define the minimal martingale measure as an equivalent martingale measure $\mathbb{Q}$ such that

(1) $\tilde{S}_t$ is a martingale under $\mathbb{Q}$,

(2) $\mathbb{P} = \mathbb{Q}$ on $\mathcal{F}_0$, and

(3) any square integrable $\mathbb{P}$-martingales orthogonal to M under $\mathbb{P}$ remains a martingale under $\mathbb{Q}$.

They also show that the minimal martingale measure $\mathbb{Q}$ is uniquely defined and exists if and only if

$$G_t = \exp\left(- \int_0^t \alpha_s dM_s - \frac{1}{2} \int_0^t \alpha_s^2 d\langle \tilde{S} \rangle_s \right) \tag{3.12}$$

is a square-integrable martingale under $\mathbb{P}$. In that case $\mathbb{Q}$ is given by:

$$\frac{d\mathbb{Q}}{d\mathbb{P}} = G_T \tag{3.13}$$

Colwell and Elliott (1993) show that the Radon-Nikodym density process G_t for Markovian models is given by

$$G_t(x_0) = 1 - \int_0^t G_{s^-}(x_0) g_s dW_s - \int_0^t \int_{\mathbb{R}} G_{s^-}(x_0)[1 - h_s]\tilde{\nu}(ds, dy) \qquad (3.14)$$

In general, g_t and h_t are suitably chosen so that $e^{-rt}S_t$ is a $\mathbb{P}^*$-martingale; in that case $\mathbb{P}^*$ is called an equivalent martingale measure. Now $e^{-rt}S_t$ is a $\mathbb{P}^*$-martingale if and only if

$$\mu + \lambda \mathbb{E}U_1 - r = \sigma g_t + \lambda \int_{\mathbb{R}} y[1 - h_t]m(dy) \qquad (3.15)$$

where m is denoted by the law of the random variable U_j, g_t is interpreted as the market price of diffusion risk, and $1 - h_t$ as the market price of jump risk.

There are infinite numbers of ways to select g_t and h_t satisfying the above Equation (3.15) for finding an equivalent martingale measure. In other words, changing of measure is "adjusting" market price of risk embedded in the underlying assets. Note that the market price of risk is the expected excess return per unit risk over the risk-free rate. It measures the trade-offs between risk and return. Colwell & Elliott (1993) show that the risk minimizing strategy leads to the following selection of market price of risk and the corresponding Radon-Nikodym derivative process

$$\begin{aligned} g &= \frac{(\mu + \lambda \mathbb{E}U_1 - r)\sigma}{\sigma^2 + \lambda \mathbb{E}U_1^2} \\ 1 - h &= \frac{(\mu + \lambda \mathbb{E}U_1 - r)U_1}{\sigma^2 + \lambda \mathbb{E}U_1^2} \qquad (3.16) \\ G_t &= 1 - \int_0^t G_{s^-} \frac{\mu + \lambda \mathbb{E}U_1 - r}{e^{-rs}S_{s^-}(\sigma^2 + \lambda \mathbb{E}U_1^2)} dM_s \end{aligned}$$

Remark 3.3.1. The market prices of risks in Merton's model, obtained by Colwell and Elliott (1993) are given by

$$\begin{cases} \quad g & = \frac{(\mu+\lambda\mathbb{E}U_1-r)}{\sigma} \\ 1-h & = 0 \end{cases} \tag{3.17}$$

As mentioned previously, the market price of jump risk in Merton's model is zero, and Merton (1976) only considers the diffusion risk inherent in risky asset's uncertainty. Comparing two sets of market prices of risks in Equation (3.16) and (3.17), we conclude that the measure under which Merton's option prices is derived and the minimal martingale measure are different. In the simple model we discuss in the last section of this Chapter, it is obvious that the market prices of diffusion and jump risks are all zero ($g = 1-h = 0$), since the left side of Equation (3.15) is zero. In other words, in the risk neutral world, the expected return of any risky asset is the risk-free rate.

Remark 3.3.2. Another feasible solution for the Equation (3.15) is

$$\begin{cases} \quad g & = \frac{(\mu-r)}{\sigma} \\ 1-h & = 1 \end{cases} \tag{3.18}$$

However this combination of market prices of risks makes the Radon-Nikodym derivative become zero, which leads a null $\mathbb{Q}$-measure. Hence another condition for the existence of equivalent martingale measures is $h > 0$, and we discuss some restrictions on the minimal martingale measure imposed by the condition.

Provided that there exists an equivalent martingale measure, we will consider the characteristics of the stock price dynamics under the new measure.

Hence by Girsanov's theorem, $W_t^* = W_t + g \cdot t$ follows a standard Brownian motion on the space $(\Omega, \mathcal{A}, \mathbb{Q})$. Zhang (1997) shows, under the transformation, that the martingale distribution of the jump size and martingale jump intensity are given by

(i) U_j are iid and $d\mathbb{Q}_{U_1}(x) = \frac{1-\eta x}{1-\eta \mathbb{E} U_1} d\mathbb{P}_{U_1}(x)$

(ii) N_t^* is a Poisson process with intensity $\lambda^* = \lambda(1 - \eta \mathbb{E} U_1)$,

where

$$\eta \equiv \frac{\mu + \lambda \mathbb{E} U_1 - r}{\sigma^2 + \lambda \mathbb{E} U_1^2}.$$

The explicit solution G_t of the above equation obtained by Zhang (1994) is given by

$$G_t = \exp(-\eta \sigma W_t - \frac{1}{2}\eta^2 \sigma^2 t) \prod_{j=1}^{N_t} (1 - \eta U_j) e^{\eta \lambda t \mathbb{E} U_1} \tag{3.19}$$

Positivity of G_t leads to the inequality $1 - \eta U_1 > 0$. If $-1 \leq \eta \leq 0$, then $1 - \eta U_1 > 0$ almost surely. If $\eta > 0$, with the assumption that $1 + U_j$ follows a log-normal distribution with mean m and variance δ^2, then $\mathbb{P}\{1 - \eta U_j \leq 0\} = \mathbb{P}\{U_j \geq \frac{1}{\eta}\} = 1 - \Phi\left(\frac{\log(1+1/\eta) - m}{\delta}\right)$. Thus if $\eta > 0$, then the probability $\mathbb{P}\{1 - \eta U_j \leq 0\}$ would have a positive value and so the equivalent probability measure might not be defined.

3.4 Valuation of a European option under minimal martingale measure

In this sub-section, we derive the dynamics of the stock price in the presence of jumps under the minimal martingale measure $\mathbb{Q}$, whose properties are discussed earlier in the previous section. Next we propose an analytical formula for a European option price under the minimal martingale measure, which we think is new in the literature.

Now, by changing of measure with the transformation, $dW_t^* = dW_t + \eta\sigma dt$, and imposing the transformation on the Equation (3.1), we obtain

$$dS_t = S_t(\mu - \eta\sigma^2)dt + \sigma S_t dW_t^* + S_t d\Big(\sum_{j=1}^{N_t^*} U_j\Big). \tag{3.20}$$

Let $\mu^* = \mu - \eta\sigma^2$. The solution of S_t is given by:

$$S_t = S_0 \exp\Big[(\mu^* - \sigma^2/2)t + \sigma W_t^*\Big]\Big(\prod_{j=1}^{N_t^*}(1 + U_j)\Big). \tag{3.21}$$

Let $\tilde{S}_t = e^{-rt}S_t$. Now

$$\begin{aligned}
\mathbb{E}^{\mathbb{Q}}(\tilde{S}_t|\mathcal{F}_s) &= \tilde{S}_s\mathbb{E}^{\mathbb{Q}}\Bigg(\exp\Big[(\mu^* - r - \sigma^2/2)(t-s) + \sigma(W_t^* - W_s^*)\Big]\prod_{j=N_s^*+1}^{N_t}(1+U_j)|\mathcal{F}_s\Bigg)\\
&= \tilde{S}_s\mathbb{E}^{\mathbb{Q}}\Bigg(\exp\Big[(\mu^* - r - \sigma^2/2)(t-s) + \sigma(W_t^* - W_s^*)\Big]\prod_{j=1}^{N_t^* - N_s^*}(1+U_{N_s+j})\Bigg)\\
&= \tilde{S}_s \exp\Big[(\mu^* - r)(t-s)\Big]\mathbb{E}^{\mathbb{Q}}\Bigg(\prod_{j=N_s+1}^{N_t^*}(1 + U_j)\Bigg)\\
&= \tilde{S}_s\ \exp\Big[(\mu^* - r)(t-s)\Big]\ \exp\Big[\lambda^*(t-s)\mathbb{E}^{\mathbb{Q}}U_1\Big]\\
&= \tilde{S}_s\ \exp\Big[(\mu^* - r + \lambda^*\mathbb{E}^{\mathbb{Q}}U_1)(t-s)\Big]
\end{aligned}$$

where $\lambda^* = \lambda(1-\eta\mathbb{E}U_1)$. From the above equation, we can conclude that $(\tilde{S}_s)$ is a martingale if and only if $\mu^* - r + \lambda^*\mathbb{E}^{\mathbb{Q}}U_1 = 0$. It is not hard to show that: $\mu^* - r + \lambda^*\mathbb{E}^{\mathbb{Q}}U_1 = (\mu - \eta\sigma^2) - r + \lambda(1-\eta\mathbb{E}U_1)\mathbb{E}^{\mathbb{Q}}U_1 = \mu - r + \lambda\mathbb{E}U_1 - \eta(\sigma^2 + \lambda\mathbb{E}U_1^2) = 0$, where

$$\begin{aligned}
\mathbb{E}^{\mathbb{Q}}U_1 &= \int_{\mathbb{R}} x \; d\mathbb{Q}_{U_1}(x) \\
&= \int_{\mathbb{R}} \frac{x(1-\eta x)}{1-\eta\mathbb{E}U_1} d\mathbb{P}_{U_1}(x) \\
&= \frac{1}{1-\eta\mathbb{E}U_1} \int_{\mathbb{R}} x \; d\mathbb{P}_{U_1}(x) - \eta \int_{\mathbb{R}} x^2 \; d\mathbb{P}_{U_1}(x) \\
&= \frac{\mathbb{E}U_1 - \eta\mathbb{E}U_1^2}{1-\eta\mathbb{E}U_1}.
\end{aligned} \tag{3.22}$$

Thus, we verify that $(\tilde{S}_s)$ is a martingale.

Now, the price process, Equation (3.20), can be rewritten by

$$dS_t = S_t(r - \lambda^*\mathbb{E}^{\mathbb{Q}}U_1)dt + \sigma S_t dW_t^* + S_t d\Big(\sum_{j=1}^{N_t^*} U_j\Big). \tag{3.23}$$

It is obvious that if we let $\eta = 0$, then $\mathbb{Q} = \mathbb{P}, \lambda^* = \lambda, W_t^* = W_t$, and $N_t^* = N_t$. This is the situation for which the actual measure is the risk-neutral measure. We call this a simple case and discuss this case in the end of this Chapter.

Now we define a European option price. As Pham (1997) points out, each equivalent martingale measure (in this case, the minimal martingale measure $\mathbb{Q}$) defines an admissible price of an option in the framework of Harrison and Kreps (1979) and Harrison and Pliska (1981). Suppose h is a real, twice differentiable payoff function, for $t > T$, the price of European option is given by

$$v(t) = \mathbb{E}^{\mathbb{Q}}[e^{-r(T-t)}h(S_T)|\mathcal{F}_t], \tag{3.24}$$

where $h(S_T) = (S_T - K)^+$ for a call, and $h(S_T) = (K - S_T)^+$ for a put.

Now define $v(t) = F(t, x)$, then

$$\begin{aligned}
F(t,x) &= \mathbb{E}^{\mathbb{Q}}\Big[e^{-r(T-t)}\ h\Big(xe^{(\mu^*-\sigma^2/2)(T-t)+\sigma W^*_{T-t}} \prod_{j=1}^{N^*_{T-t}} (1+U_j)\Big)\Big] \\
&= \mathbb{E}^{\mathbb{Q}}\Big[e^{-r(T-t)}\ h\Big(xe^{(r-\lambda^*\mathbb{E}^{\mathbb{Q}}U_1-\sigma^2/2)(T-t)+\sigma W^*_{T-t}} \prod_{j=1}^{N^*_{T-t}} (1+U_j)\Big)\Big] \\
&= \mathbb{E}^{\mathbb{Q}}\Big[F_0\big(t, xe^{-\lambda^*(T-t)\mathbb{E}^{\mathbb{Q}}U_1} \prod_{j=1}^{N^*_{T-t}} (1+U_j)\big)\Big] \\
&= \sum_{n=0}^{\infty} \frac{e^{-\lambda^*(T-t)}(\lambda^*)^n(T-t)^n}{n!}\ \mathbb{E}^{\mathbb{Q}}\Big[F_0\big(t, xe^{-\lambda^*(T-t)\mathbb{E}^{\mathbb{Q}}U_1} \prod_{j=1}^{n}(1+U_j)\big)\Big],
\end{aligned} \tag{3.25}$$

where $F_0(t, x)$ is the Black-Scholes option pricing formula for the no-jump case defined by:

$$F_0(t,x) = \mathbb{E}^{\mathbb{Q}}\Big[e^{-r(T-t)}\ h\Big(xe^{(r-\sigma^2/2)(T-t)+\sigma W^*_{T-t}}\Big)\Big]$$

Notice that if we set $\eta = 0$, then the above equation is equivalent to the option pricing formula of Merton (1976).

Proposition 3.4.1. *If we assume that the random variable* $\ln(1 + U_j)$ *follows a normal distribution with mean* m *and variance* δ^2 *under* $\mathbb{Q}$*, then* $F(t, x)$ *can be computed analytically as follows.*

$$F(t,x) = \sum_{n=0}^{\infty} \frac{e^{-\lambda'(T-t)}(\lambda'(T-t))^n}{n!}\ \Big\{e^{-r_n(T-t)}K\Phi(-d_2) - x\Phi(-d_1)\Big\}, \tag{3.26}$$

where d_1, d_2 *are denoted by:*

$$\begin{aligned} -d_2 &= \frac{\ln(K/x) - (r_n - \frac{1}{2}\nu_n^2)(T-t)}{\nu_n\sqrt{T-t}} \\ -d_1 &= -d_2 + \nu_n\sqrt{T-t}, \end{aligned} \tag{3.27}$$

where $\lambda^{'} \equiv \lambda^*(1+\mathbb{E}^{\mathbb{Q}}U_1)$, $r_n \equiv r - \lambda^*\mathbb{E}^{\mathbb{Q}}U_1 + n\gamma/(T-t)$, *and* $\nu_n^2 \equiv \sigma^2 + n\delta^2/(T-t)$. *Note that* $\gamma \equiv \ln \mathbb{E}^{\mathbb{Q}}(1+U_1) = \frac{\delta^2}{2} + m$.

Proof. Without loss of generality, we here consider European put price with $t = 0$, $x = S_0$, and exercise price, K.

$$\begin{aligned} &\mathbb{E}^{\mathbb{Q}}(F_0\big(0, S_0 e^{-\lambda^*\mathbb{E}^{\mathbb{Q}}U_1 T}\prod_{j=1}^{n}(1+U_j)\big)) \\ &\quad = \mathbb{E}^{\mathbb{Q}}[e^{-rT}(K-S_T)^+ | n\ jumps] \quad \text{(By recalling the process of B-S model)} \\ &\quad = \mathbb{E}^{\mathbb{Q}}\left(e^{-rT}\left[K - S_0 e^{-\lambda^*\mathbb{E}^{\mathbb{Q}}U_1 T}\prod_{j=1}^{n}(1+U_j)\exp\left(\sigma W_T + (r - \frac{1}{2}\sigma^2)T\right)\right]^+\right) \\ &\quad = \mathbb{E}^{\mathbb{Q}}\left(e^{-rT}\left[K - S_0\exp\left(\sigma W_T + (r - \lambda^*\mathbb{E}^{\mathbb{Q}}U_1 - \frac{1}{2}\sigma^2)T + \sum_{j=1}^{n}\ln(1+U_j)\right)\right]^+\right) \\ &\quad = \mathbb{E}^{\mathbb{Q}}\left(\left[e^{-rT}K - S_0\exp\left(\sigma W_T - (\lambda^*\mathbb{E}^{\mathbb{Q}}U_1 + \frac{1}{2}\sigma^2)T + \sum_{j=1}^{n}\ln(1+U_j)\right)\right]^+\right) \\ &\quad = \mathbb{E}^{\mathbb{Q}}\left(\left[e^{-rT}K - S_0 e^{z\sqrt{\sigma^2 T + n\delta^2} + nm - \lambda^* k^* T - \frac{1}{2}\sigma^2 T}\right]^+\right), \end{aligned}$$

where z follows a standard Gaussian law $N(0,1)$, $k^* = \mathbb{E}^{\mathbb{Q}}U_1$, and m, and δ^2 is mean and the variance of $\ln(1+U_1)$, respectively.

Note that

$$
\begin{aligned}
& e^{-rT}K - S_0 e^{z\sqrt{\sigma^2 T + n\delta^2} + nm - \lambda^* k^* T - \frac{1}{2}\sigma^2 T} \geq 0 \\
\Rightarrow \quad & z \leq \ln(K/S_0) - (r - \lambda^* k^*)T - nm + \frac{1}{2}\sigma^2 T \\
\Rightarrow \quad & z \leq \ln(K/S_0) - \Big(r - \lambda^* k^* + \frac{n\gamma}{T}\Big)T + \frac{1}{2}\Big(\sigma^2 + \frac{n\delta^2}{T}\Big)T \\
\Rightarrow \quad & z \leq \ln(K/S_0) - (r_n - \frac{1}{2}\nu_n^2)T
\end{aligned}
$$

where

$$
\begin{aligned}
r_n &= r - \lambda^* k^* + \frac{n\gamma}{T} \\
\nu_n^2 &= \sigma^2 + \frac{n\delta^2}{T}
\end{aligned}
$$

Now simple algebra with the previous result gives:

$$
\begin{aligned}
& F(0, S_0) \\
& \quad = \sum_{n=0}^{\infty} \frac{e^{-\lambda^* T}(\lambda^* T)^n}{n!} \mathbb{E}\left(e^{-rT}K - S_0 e^{z\sqrt{\sigma^2 T + n\delta^2} + nm - \lambda^* k^* T - \frac{1}{2}\sigma^2 T} I_{\{g + d_2 \leq 0\}} \right) \\
& \quad = \sum_{n=0}^{\infty} \frac{e^{-\lambda^* T}(\lambda^* T)^n}{n!} \int_{-\infty}^{-d_2} \left(e^{-rT}K - S_0 e^{y\sqrt{\sigma^2 T + n\delta^2} + nm - \lambda^* k^* T - \frac{1}{2}\sigma^2 T} \right) \frac{e^{-y^2/2}}{\sqrt{2\pi}}\, dy \\
& \quad = \sum_{n=0}^{\infty} \frac{e^{-\lambda^* T}(\lambda^* T)^n}{n!} \Bigg\{ e^{-rT}K \int_{-\infty}^{-d_2} \frac{e^{-y^2/2}}{\sqrt{2\pi}}\, dy \\
& \qquad\qquad\qquad - S_0 \int_{-\infty}^{-d_2} e^{y\sqrt{\sigma^2 T + n\delta^2} + nm - \lambda^* k^* T - \frac{1}{2}\sigma^2 T} \ \frac{e^{-y^2/2}}{\sqrt{2\pi}}\, dy \Bigg\} \\
& \quad = \sum_{n=0}^{\infty} \frac{e^{-\lambda^* T}(\lambda^* T)^n}{n!} \Bigg\{ e^{n\gamma - \lambda^* k^* T}\ e^{-r_n T}K\ \Phi(-d_2) \\
& \qquad\qquad\qquad - S_0 \int_{-\infty}^{-d_2} e^{y\sqrt{\sigma^2 T + n\delta^2} + nm - \lambda^* k^* T - \frac{1}{2}\sigma^2 T} \ \frac{e^{-y^2/2}}{\sqrt{2\pi}}\, dy \Bigg\}
\end{aligned} \tag{3.28}
$$

Evaluation of the second integral by changing of variable $\tilde{y} = y - \nu_n\sqrt{T}$ gives

$$\begin{aligned}
&\int_{-\infty}^{-d_2} e^{y\sqrt{\sigma^2T+n\delta^2}+nm-\lambda^*k^*T-\frac{1}{2}\sigma^2T} \ \frac{e^{-y^2/2}}{\sqrt{2\pi}}\, dy \\
&\qquad = \int_{-\infty}^{-d_1} e^{\tilde{y}\sqrt{\sigma^2T+n\delta^2}+(\sigma^2T+n\delta^2)+nm-\lambda^*k^*T-\frac{1}{2}\sigma^2T} \ \frac{e^{-(\tilde{y}+\sqrt{\sigma^2T+n\delta^2})^2/2}}{\sqrt{2\pi}}\, d\tilde{y} \\
&\qquad = e^{n\delta^2/2+nm-\lambda^*k^*T} \int_{-\infty}^{-d_1} \frac{e^{-\tilde{y}^2/2}}{\sqrt{2\pi}}\, d\tilde{y} \\
&\qquad = e^{n\gamma-\lambda^*k^*T} \int_{-\infty}^{-d_1} \frac{e^{-\tilde{y}^2/2}}{\sqrt{2\pi}}\, d\tilde{y} \\
&\qquad = e^{n\gamma-\lambda^*k^*T}\ \Phi(-d_1)
\end{aligned} \tag{3.29}$$

Combining Equation (3.29) into the Equation (3.28), we obtain:

$$\begin{aligned}
&F(0,S_0) \\
&\quad = \sum_{n=0}^{\infty} \frac{e^{-\lambda^*T}(\lambda^*T)^n}{n!} \left\{ e^{n\gamma-\lambda^*k^*T}\ e^{-r_nT}K\ \Phi(-d_2) - S_0\ e^{n\gamma-\lambda^*k^*T}\ \Phi(-d_1) \right\} \\
&\quad = \sum_{n=0}^{\infty} \frac{e^{-\lambda^{'}T}(\lambda^{'}T)^n}{n!} \left\{ e^{-r_nT}K\Phi(-d_2) - S_0\ \Phi(-d_1) \right\}.
\end{aligned} \tag{3.30}$$

where $\lambda^{'} \equiv \lambda^*(1 + \mathbb{E}^{\mathbb{Q}}U_1)$. □

The formula of option prices is very similar to Merton (1976). The only unknown parameter is η. Note that if we set $\eta = 0$, the above Equation (3.26) is equivalent to what Merton (1976) derives.

3.5 A simple case

It is clear as shown in the previous section that the discounted stock price process is a martingale under the actual measure $\mathbb{P}$, if and only if $\mu =$

$r - \lambda\mathbb{E}U_1$. Thus we regard the actual measure with the condition $\mu = r - \lambda\mathbb{E}U_1$ as the risk neutral measure in this market. Thus the process of stock price, Equation (3.1), can be rewritten by

$$\frac{dS_t}{S_{t^-}} = (r - \lambda\mathbb{E}U_1)dt + \sigma dW_t + d\left(\sum_{j=1}^{N_t} U_j\right) \tag{3.31}$$

The above stochastic differential equation is a special case of Equation (3.23). Thus from Equations (3.23) and (3.30), the European option price obtained in this simple case is the same as Merton's (1976). Although the market prices of risks are different as we see earlier in this Chapter, the models render the same formula for a European option price.

Chapter 4

Pricing American Option Under Jump-Diffusion using Early Exercise Premium Representation

In this chapter, we derive an American option pricing formula as an early exercise premium representation under jump-diffusion processes. We decompose the value of an American option into that of a European counter part and an early exercise premium. The technique is based on the methodology of Kim (1990) who introduced the decomposition method in the pure diffusion model. In deriving the value of an American option, we use the minimal martingale measure $\mathbb{Q}$ whose characteristics are discussed in the previous chapter.

In section 1, we provide the alternative derivation of the price of an American put option on a stock with no dividend under jump-diffusion processes. In section 2, the price of American call with dividends is proposed. Finally, section 3 presents the valuation formulas of American puts with various distributions for jump sizes.

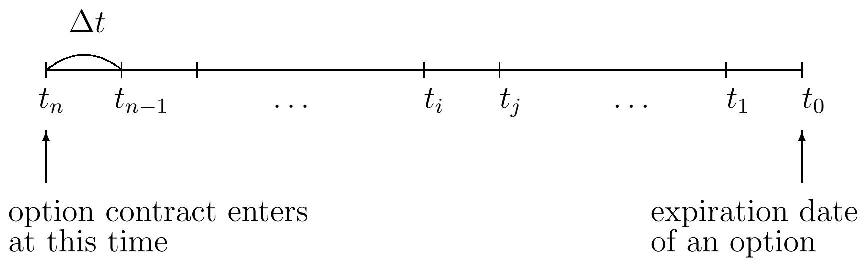


Figure 4.1: Discrete time space

4.1 Valuation of American puts

We consider in this section an American put option written on a non-dividend paying stock in the presence of jumps. As derived earlier in the previous chapter, the dynamics of the stock price is given by

$$\frac{dS_t}{S_{t^-}} = (r - \lambda^* \mathbb{E}^{\mathbb{Q}} U_1)dt + \sigma dW_t^* + d\Big(\sum_{j=1}^{N_t^*} U_j\Big),$$

under the minimal martingale measure $\mathbb{Q}$. To approximate the American option price, we first discretize the continuous time model following the approach of Kim (1990) and then consider the limiting behavior of the discrete model. We assume that the American put can be exercised at a finite number of points in time denoted by t_k, $k = 0, 1, \ldots, n$, where $t_k - t_{k-1} = \Delta t$ for all k. We suppose that the option enters a contract at time t_n and expires at t_0. The time to maturity, τ, of this put is $n\Delta t$. Define S_k, and B_k as the stock price and critical stock price with time to maturity of $k\Delta t$, respectively. Thus S_n is the initial stock price, S_0 is the stock price at expiration date, and B_0 is the critical stock price at the expiration date, which is equivalent to the exercise price.

We define $\psi(S_j,(i-j)\Delta t; S_{i^-})$ for $i > j$ as the *transition density function* which denotes the probability density function of the asset price S_j at time t_j provided that the asset price at t_i is S_{i^-} under the minimal martingale measure, $\mathbb{Q}$. We assume that the sample path of a stock price is right continuous and left limit, and any size and any numbers of jumps can occur on the time interval $[t_i, t_j]$. We suppress the superscript on the time index (i.e., i^-) for a simple notation.

With this framework, we show that the value of American put can be decomposed into that of European put and the early exercise premium. Define $V(x,\tau)$ as the value of American put when the initial stock price is x and the time to maturity is τ. Define also $p(x,\tau)$, $c(x,\tau)$ as that of European put and of European call respectively. Assume also that we consider only *live* American options in a sense that it is not optimal to initially exercise the options immediately.

Proposition 4.1.1. *The price of an American put with a maturity of $\tau = n\Delta t$ is given by:*

$$\begin{aligned}V(S_n, n\Delta t) &= p(S_n, n\Delta t)\\ &\quad + \sum_{k=1}^{n-1} e^{-(n-k)r\Delta t} \int_0^{B_k} [K - S_k - V(S_k, k\Delta t)]\ \psi(S_k,(n-k)\Delta t; S_n)\ dS_k\end{aligned} \tag{4.1}$$

Proof. We will prove the assertion by using mathematical induction. The value of American put with a maturity of Δt, $V(S_1, \Delta t)$, is identical to that of an equivalent European put, $p(S_1, \Delta t)$, since exercising the option at the current time is not optimal and there is no exercising opportunity before the maturity date. Thus,

$$V(S_1, \Delta t) = p(S_1, \Delta t)$$

Next, the value of American put at $\tau = 2\Delta t$, $V(S_2, 2\Delta t)$, is given by

$$\begin{aligned} V(S_2,&2\Delta t) \\ &= \int_{B_1}^{\infty} e^{-r\Delta t} V(S_1, \Delta t)\psi(S_1, \Delta t; S_2)\, dS_1 \\ &\quad + \int_0^{B_1} e^{-r\Delta t}(K - S_1)\psi(S_1, \Delta t; S_2)\, dS_1 \\ &= \int_0^{\infty} e^{-r\Delta t} V(S_1, \Delta t)\psi(S_1, \Delta t; S_2)\, dS_1 \\ &\quad + \int_0^{B_1} e^{-r\Delta t}[K - S_1 - V(S_1, \Delta t)]\psi(S_1, \Delta t; S_2)\, dS_1 \\ &= p(S_2, 2\Delta t) \\ &\quad + \int_0^{B_1} e^{-r\Delta t}[K - S_1 - V(S_1, \Delta t)]\psi(S_1, \Delta t; S_2)\, dS_1 \end{aligned}$$

Assume that the value of $V(S_n, n\Delta t)$ is given by

$$\begin{aligned} V(S_n,&n\Delta t) = p(S_n, n\Delta t) \\ &+ \sum_{k=1}^{n-1} e^{-(n-k)r\Delta t} \int_0^{B_k} [K - S_k - V(S_k, k\Delta t)]\ \psi(S_k, (n-k)\Delta t; S_n)\, dS_k \end{aligned} \tag{4.2}$$

Now consider the American put price, $V(S_{n+1}, (n+1)\Delta t)$ at $\tau = (n+1)\Delta t$, and it is given by

$$
\begin{aligned}
V(S_{n+1},&(n+1)\Delta t) \\
&= \int_{B_n}^{\infty} e^{-r\Delta t} V(S_n, n\Delta t)\psi(S_n, \Delta t; S_{n+1})\, dS_n \\
&\quad + \int_0^{B_n} e^{-r\Delta t}(K - S_n)\psi(S_n, \Delta t; S_{n+1})\, dS_n \\
&= \int_0^{\infty} e^{-r\Delta t} V(S_n, n\Delta t)\psi(S_n, \Delta t; S_{n+1})\, dS_n \\
&\quad + \int_0^{B_n} e^{-r\Delta t}[K - S_n - V(S_n, n\Delta t)]\psi(S_n, \Delta t; S_{n+1})\, dS_n
\end{aligned}
$$

By replacing $V(S_n, n\Delta t)$ in the first term of the last equation with the right term of Equation (4.2), we obtain the following:

$$
\begin{aligned}
&V(S_{n+1}, (n+1)\Delta t) \\
&\quad = \int_0^{\infty} e^{-r\Delta t} p(S_n, n\Delta t)\ \psi(S_n, \Delta t; S_{n+1})\, dS_n \\
&\qquad + \int_0^{\infty} e^{-r\Delta t} \left[\sum_{k=1}^{n-1} e^{-(n-k)r\Delta t} \int_0^{B_k} [K - S_k - V(S_k, k\Delta t)]\ \psi(S_k, (n-k)\Delta t; S_n)\, dS_k \right] \\
&\qquad \psi(S_n, \Delta t; S_{n+1})\, dS_n \\
&\qquad + \int_0^{B_n} e^{-r\Delta t}[K - S_n - V(S_n, n\Delta t)]\psi(S_n, \Delta t; S_{n+1})\, dS_n \\
&\quad = p(S_{n+1}, (n+1)\Delta t) \\
&\qquad + \sum_{k=1}^{n-1} e^{-(n-k+1)r\Delta t} \int_0^{B_k} [K - S_k - V(S_k, k\Delta t)]\ \psi(S_k, (n-k+1)\Delta t; S_{n+1})\, dS_k \\
&\qquad + \int_0^{B_n} e^{-r\Delta t}[K - S_n - V(S_n, n\Delta t)]\psi(S_n, \Delta t; S_{n+1})\, dS_n \\
&\quad = p(S_{n+1}, (n+1)\Delta t) \\
&\qquad\quad + \sum_{k=1}^{n} e^{-(n+1-k)r\Delta t} \int_0^{B_k} [K - S_k - V(S_k, k\Delta t)]\ \psi(S_k, (n+1-k)\Delta t; S_{n+1})\, dS_k,
\end{aligned}
$$

which gives the result. $\square$

Now, consider the value of American put with a maturity of Δt, $V(S_1, \Delta t)$. The value of $V(S_1, \Delta t)$ is identical to an equivalent European put, $p(S_1, \Delta t)$, since there is no early exercise opportunity before the maturity date. Thus $V(S_1, \Delta t) = p(S_1, \Delta t)$.

Next, consider the value of American put with a maturity of $2\Delta t$, $V(S_2, 2\Delta t)$. Using the previous proposition, we can show that

$$\begin{aligned} V(S_2, 2\Delta t) &= p(S_2, 2\Delta t) + \int_0^{B_1} e^{-r\Delta t}[K - S_1 - V(S_1, \Delta t)]\psi(S_1, \Delta t; S_2)\, dS_1 \\ &= p(S_2, 2\Delta t) + \int_0^{B_1} e^{-r\Delta t}[K - S_1 - p(S_1, \Delta t)]\psi(S_1, \Delta t; S_2)\, dS_1 \\ &= p(S_2, 2\Delta t) + e^{-r\Delta t}\int_0^{B_1} (1 - e^{-r\Delta t})K\psi(S_1, \Delta t; S_2)\, dS_1 \\ &\quad - e^{-r\Delta t}\int_0^{B_1} c(S_1, \Delta t)\psi(S_1, \Delta t; S_2)\, dS_1. \end{aligned} \tag{4.3}$$

The last equality holds by the put-call parity which is characterized by the following form:

$$c(S_1, \Delta t) + Ke^{-r\Delta t} = p(S_1, \Delta t) + S_1 \tag{4.4}$$

Note that the value of American put in Equation (4.3) contains a price of European call price. However as the following proposition shows, the integral term having European call option prices as its integrand vanishes to zero with a high speed as Δt goes to zero.

Proposition 4.1.2. *When the jump size has a log-normal distribution under* $\mathbb{Q}$ *(i.e.,* $\ln(1+U_1)$ *follows a normal distribution with mean of* m *and standard deviation of* δ*.), the third term of* $V(S_2, 2\Delta t)$ *of Equation (4.3) is of order higher than* Δt*, which we denote* $o(\Delta t)$*, and the value of the American put with a maturity of* $2\Delta t$ *is given by*

$$V(S_2, 2\Delta t) = p(S_2, 2\Delta t) + e^{-r\Delta t} \int_0^{B_1} (1 - e^{-r\Delta t}) K \psi(S_1, \Delta t; S_2)\, dS_1 + o(\Delta t) \tag{4.5}$$

Proof. Since $c(S_1, \Delta t) < c(B_1, \Delta t)$ for $0 < S_1 < B_1$, the following inequality holds:

$$\int_0^{B_1} c(S_1, \Delta t)\ \psi(S_1, \Delta t; S_2)\, dS_1 \\ < \ c(B_1, \Delta t) \int_0^{B_1} \ \psi(S_1, \Delta t; S_2)\, dS_1$$

We show that the left side of the above inequality is $o(\Delta t)$ by proving that the order of each term of the right side of the inequlity is equal to or higher than Δt, and thus the multiplication of the two terms is of order higher than Δt. First we show that $\int_0^{B_1} \ \psi(S_1, \Delta t; S_2)\, dS_1$ is of order Δt or higher. By expressing the transition density function explicitly, we obtain the following:

$$
\begin{aligned}
&\int_0^{B_1} \psi(S_1, \Delta t; S_2)\, dS_1 \\
&\qquad = \int_0^{B_1} \sum_{n=0}^{\infty} e^{-\lambda^* \Delta t} \frac{(\lambda^* \Delta t)^n}{n!} \psi(S_1, \Delta t; S_2 | n\ jumps)\, dS_1 \\
&\qquad = \sum_{n=0}^{\infty} e^{-\lambda^* \Delta t} \frac{(\lambda^* \Delta t)^n}{n!} \int_0^{B_1} \psi(S_1, \Delta t; S_2 | n\ jumps)\, dS_1 \\
&\qquad = \sum_{n=0}^{\infty} e^{-\lambda^* \Delta t} \frac{(\lambda^* \Delta t)^n}{n!} \cdot \\
&\qquad\qquad \cdot \mathbb{Q} \left\{ S_2 \exp\left(\sigma W_{\Delta t}^* + (r - \lambda^* k^* - \frac{1}{2}\sigma^2)\Delta t \right) \prod_{j=1}^{n} (1 + U_j) < B_1 \right\} \\
&\qquad = \sum_{n=0}^{\infty} e^{-\lambda^* \Delta t} \frac{(\lambda^* \Delta t)^n}{n!} \cdot \\
&\qquad\qquad \cdot \mathbb{Q} \left\{ S_2 \exp\left(\sigma W_{\Delta t}^* + (r - \lambda^* k^* - \frac{1}{2}\sigma^2)\Delta t + \sum_{j=1}^{n} \ln(1 + U_j) \right) < B_1 \right\} \\
&\qquad = \sum_{n=0}^{\infty} e^{-\lambda^* \Delta t} \frac{(\lambda^* \Delta t)^n}{n!} \cdot \\
&\qquad\qquad \cdot \mathbb{Q} \left\{ \sigma W_{\Delta t}^* + \sum_{j=1}^{n} \ln(1 + U_j) < \ln(B_1/S_2) - (r - \lambda^* k^* - \frac{1}{2}\sigma^2)\Delta t \right\}
\end{aligned}
$$

where $k^* = \mathbb{E}^{\mathbb{Q}} U_1$. Note that $\mathbb{Q}$ is a minimal martingale measure. If the jump size $1 + U_1$ follows a log-normal distribution (i.e, $\ln(1 + U_j)$ follows $\mathcal{N}(m, \delta^2)$)

under $\mathbb{Q}$, the above equation can be reformulated by

$$\begin{aligned}
&\int_0^{B_1} \psi(S_1, \Delta t; S_2)\, dS_1 \\
&\quad = \sum_{n=0}^{\infty} e^{-\lambda^* \Delta t} \frac{(\lambda^* \Delta t)^n}{n!} \mathbb{Q}\left\{ \xi < \frac{\ln(B_1/S_2) - (r - \lambda^* k^* - \frac{1}{2}\sigma^2)\Delta t - nm}{\sqrt{\sigma^2 \Delta t + n\delta^2}} \right\} \\
&\quad = \sum_{n=0}^{\infty} e^{-\lambda^* \Delta t} \frac{(\lambda^* \Delta t)^n}{n!}\ \Phi\left(\frac{\ln(B_1/S_2) - (r - \lambda^* k^* + \frac{\ln(1+k^*)^n}{\Delta t} - \frac{1}{2}\nu_n^2)\Delta t}{\nu_n \sqrt{\Delta t}} \right) \\
&\quad \leq \sum_{n=1}^{\infty} e^{-\lambda^* \Delta t} \frac{(\lambda^* \Delta t)^n}{n!} + e^{-\lambda^* \Delta t} \Phi\left(\frac{\ln(B_1/S_2) - (r - \frac{1}{2}\sigma^2)\Delta t}{\sigma \sqrt{\Delta t}} \right)
\end{aligned}$$

where $\nu_n^2 = \sigma^2 + n\delta^2/\Delta t$, and Φ is the standard normal distribution function. The first term is of order Δt. The second term is a no-jump case, and is of order Δt or higher, which is proved by Kim (1990). Thus $\int_0^{B_1} \psi(S_1, \Delta t; S_2)\, dS_1$ is of order Δt or higher.

Next we show that $c(B_1, \Delta t)$ is of order Δt or higher. Similar to the method applied in the proposition 3.4.1 of the previous chapter, the European call price in the presence of jump can be represented as follows:

$$\begin{aligned}
c(B_1, \Delta t) &= \sum_{n=0}^{\infty} \frac{e^{-\lambda' \Delta t} (\lambda' \Delta t)^n}{n!} c_n(B_1, \Delta t) \\
&= \sum_{n=1}^{\infty} \frac{e^{-\lambda' \Delta t} (\lambda' \Delta t)^n}{n!} c_n(B_1, \Delta t) \ + \ e^{-\lambda \Delta t} c_0(B_1, \Delta t)
\end{aligned}$$

where $\lambda' = \lambda^*(1 + k^*)$. Each of the term is of order Δt or higher, so $c(B_1, \Delta t)$ is of order Δt or higher. Thus, the multiplication of two terms, whose orders are Δt or higher, is of order higher than Δt. $\square$

Theorem 4.1.3. *For the general case, when the jump size has a log-normal* *distribution under $\mathbb{Q}$, the American put price with jump-diffusion is given by:*

$$V(S_n, n\Delta t) = p(S_n, n\Delta t) + (1 - e^{-r\Delta t})K \sum_{k=1}^{n-1} \left[e^{-(n-k)r\Delta t} \int_0^{B_k} \psi(S_k, (n-k)\Delta t; S_n)\, dS_k \right] + o(\Delta t) \tag{4.6}$$

In the continuous time, the value of American put price in the presence of jump-diffusion processes is given by

$$V(S, \tau) = p(S, \tau) + rK \int_0^{\tau} e^{-r(\tau - s)} \left[\int_0^{B(s)} \psi(S_s, \tau - s; S)\, dS_s \right] ds \tag{4.7}$$

where

$$p(S, \tau) = \sum_{m=0}^{\infty} \frac{e^{-\lambda'\tau}(\lambda'\tau)^m}{m!} p_m(S, \tau), \tag{4.8}$$

$$p_m(S, \tau) = K e^{-(r - \lambda^* k^* + \frac{n\gamma}{\tau})\tau} \Phi(-d_2') - S\Phi(-d_1') \tag{4.9}$$

$$-d_{1,2}' = \frac{\ln(K/S) - (r - \lambda^* k^* + \frac{m\gamma}{\tau} \pm \frac{1}{2}\nu_m^2)\tau}{\nu_m \sqrt{\tau}} \tag{4.10}$$

and

$$\int_0^{B_s} \psi(S_s, \tau - s; S) dS_s = \sum_{m=0}^{\infty} e^{-\lambda^*(\tau - s)} \frac{(\lambda^*(\tau - s))^m}{m!} \cdot \Phi\left(\frac{\ln(B_s/S) - (r - \lambda^* k^* + \frac{m\gamma}{(\tau - s)} - \frac{1}{2}\nu_m^2)(\tau - s)}{\nu_m \sqrt{\tau - s}} \right) \tag{4.11}$$

where $\lambda' \equiv \lambda^*(1 + k^*)$, $\gamma \equiv \log(1 + k^*)$, *and* $\nu_m^2 \equiv \sigma^2 + \frac{m\delta^2}{\tau - s}$.

Proof. See the appendix. □

Remark 4.1.1. If there is no jump, the price of an American put is given by

$$V(S,\tau) = p(S,\tau) + rK \int_0^\tau e^{-r(\tau-s)}\ \Phi(-d_2)\ \ ds, \tag{4.12}$$

where

$$-d_2 = \frac{\ln(B(s)/S) - (r - \frac{1}{2}\sigma^2)(\tau - s)}{\sigma\sqrt{\tau - s}} \tag{4.13}$$

This formula is identical to those derived by Kim (1990), Jacka (1991), and Carr, Jarrow, and Myneni (1992).

Remark 4.1.2. Pham (1997) and Gukhal (2001) derive the early exercise representation of the price of an American put. When the jump risk is unpriced, the formula is as follows:

$$V(S,\tau) = p(S,\tau) + e(S,\tau) \tag{4.14}$$

where e is the early exercise premium: $e = e_1 - e_2$, with

$$\begin{aligned}
e_1(S,\tau) &= rK \int_0^\tau e^{-r(\tau-s)}\ \mathbb{P}^*\big[S_s \le B(s)\big]\ ds, \\
e_2(S,\tau) &= \lambda \mathbb{E}^* \Bigg[\int_0^\tau \int_A e^{-r(\tau-s)} \chi\Big(S_s \le B(s)\Big) \\
&\qquad \times \Big\{ V\big(S_s[1+\gamma(y)],\ \tau - s\big) - \big(K - S_s[1+\gamma(y)]\big)\Big\}\ m(dy)\ ds \Bigg],
\end{aligned}$$

where $\mathbb{P}^*$ is an equivalent martingale measure, χ is the characteristic function, $A = \{y \in \mathbb{R},\ S_s(1+\gamma(y)) > B(s)\}$, $m(dy)$ is the probability measure on $\mathbb{R}$ of the independent identically distributed random variable Y_n, also independent of N_t, and $\gamma(Y_n)_{n\in\mathbb{N}}$ are the square integrable random jump relative sizes of the stock price where $1+\gamma > 0$, Note that e_2, which is an extra term for the jump process and includes terms of the American put price recursively, vanishes in our early exercise representation.

4.2 Valuation of American calls with dividends

In this section, we derive the analytical valuation formula for an American call with dividends under the jump-diffusion processes, based on the same assumptions and methodology used in the previous section. However we consider only the case of $\eta = 0$ for the simplicity of analysis and derivation. The non-zero case is easily extended from this simple case. We assume that the dividend rate that is paid to the shareholders during an option's life time can be predicted with certainty. Provided that the stock S_t continuously pays dividends at some fixed rate α, the value of the American call is given by the following result.

Theorem 4.2.1. *The value of American call with dividends is given by*

$$\begin{aligned} V(S,\tau) = c(S,\tau) &+ \int_0^\tau \alpha S e^{-\alpha(\tau-s)} \left[\int_{B(s)}^\infty \psi_1(S_s, \tau - s; S) dS_s \right] ds \\ &- \int_0^\tau rKe^{-r(\tau-s)} \left[\int_{B(s)}^\infty \psi_2(S_s, \tau - s; S) dS_s \right] ds \end{aligned} \quad (4.15)$$

where, $c(S,\tau)$ *is the price of European counterpart and given by,*

$$c(S,\tau) = \sum_{m=0}^\infty \frac{e^{-\lambda'\tau}(\lambda'\tau)^m}{m!} c_m(S,\tau), \quad (4.16)$$

$$c_m(S,\tau) = e^{-\alpha\tau} S\Phi(d_1') - Ke^{-r_m\tau}\Phi(d_2') \quad (4.17)$$

$$d'_{1,2} = \frac{\ln(S/K) + (r_m \pm \frac{1}{2}\nu_m^2)\tau}{\nu_m \sqrt{\tau}} \tag{4.18}$$

$$\int_{B(s)}^{\infty} \psi_{1,2}(S_s, \tau - s; S) dS_s = \sum_{m=0}^{\infty} e^{-\lambda(\tau - s)} \frac{(\lambda(\tau - s))^m}{m!} \cdot \Phi\left(\frac{\ln(S_s/(B(s)) + (r_m \pm \frac{1}{2}\nu_m^2)(\tau - s)}{\nu_m \sqrt{(\tau - s)}} \right) \tag{4.19}$$

where $r_m = r - \alpha - \lambda k + \frac{m\gamma}{\tau}$.

Proof. The methodology of proof for the assertion is similar to that applied in the proof of Theorem 4.1.3. □

4.3 Valuation of American options with various distributions of jump size

In this section, we derive the analytical valuation formula for an American put under the jump-diffusion processes with various distributions of jump size, based on the same assumptions and methodology used in the previous sections. Up to now we assume that random variable $1 + U_j$ follows only the log-normal distribution. In this section, we consider the following three distributions for the random jump sizes which has already been proposed in the literature.

- Jump-to-ruin process (Merton (1976), Longstaff & Schwartz (2001))
- Bivariate jumps (Amin (1993))

- Double exponential density (Kou (2000))

We apply the above distributions to the Amin's and our proposed models and compare the option prices. As in the previous section, we assume here $\eta = 0$ without loss of generality.

4.3.1 Jump-to-ruin process

We consider a special case for the distribution of jump size where there is a positive probability of immediate ruin. In other words, if the Poisson event occurs, then the stock price goes to zero. Thus, in this case, $\prod_{j=1}^{n}(1+U_j) = 0$ for $n \neq 0$, and $\mathbb{E}U_j = -1$. Merton derives the European option price when the jump follows a jump-to-ruin process.

$$\begin{aligned} p(S,\tau) &= e^{-\lambda\tau} f^{BS}(Se^{\lambda\tau}, \tau; K, \sigma^2, r) \\ &= f^{BS}(S, \tau; K, \sigma^2, r+\lambda) \end{aligned} \tag{4.20}$$

Theorem 4.3.1. *The price of an American put with jump-to-ruin process is given by*

$$V(S,\tau) = p(S,\tau) + rK \int_0^\tau e^{-r(\tau-s)} \left[\int_0^{B(s)} \psi(S_s, \tau-s; S)\, dS_s \right] ds \tag{4.21}$$

where

$$\int_0^{B(s)} \psi(S_s, \tau-s; S)\, dS_s = e^{-\lambda(\tau-s)}\, \Phi\left(\frac{\ln(B(s)/S) - (r+\lambda-\frac{1}{2}\sigma^2)(\tau-s)}{\sigma\sqrt{\tau-s}} \right) \tag{4.22}$$

Proof. Based on the proofs of the Propositions 4.1.1, 4.1.2, and the Theorem 4.1.3, it is sufficient to show that $\int_0^{B_1} \psi(S_1, \Delta t; S_2)$ is of order Δt or higher.

Now, from the following equation:

$$\int_0^{B_1} \psi(S_1, \Delta t; S_2)\, dS_1 = e^{-\lambda \Delta t}\ \Phi\left(\frac{\ln(B_1/S_1) - (r + \lambda - \frac{1}{2}\sigma^2)\Delta t}{\sigma\sqrt{\Delta t}}\right), \quad (4.23)$$

it is not difficult to see that the right side of the equation above is of order Δt or higher. $\square$

4.3.2 Bivariate jumps

We assume here that the jump distribution is specified by

$$\log(1+U) = \begin{cases} +\xi & \text{with prob. } b, \\ -\xi & \text{with prob. } 1-b, \end{cases} \quad (4.24)$$

where b is a constant in $(0, 1)$. Assuming the jump size follows the bivariate distribution, the European put option price is given by

$$p(S,\tau) = \sum_{n=0}^{\infty} \frac{e^{-\lambda\tau}(\lambda\tau)^n}{n!} \sum_{i=0}^{n} \frac{n!}{i!(n-i)!} b^i (1-b)^{n-i}\ f^{BS}(Se^{-\lambda k\tau + (2i-n)\xi}, \tau; K, \sigma^2, r) \quad (4.25)$$

or

$$p(S,\tau) = \sum_{n=0}^{\infty} \frac{e^{-\lambda\tau}(\lambda\tau)^n}{n!} \sum_{i=0}^{n} \frac{n!}{i!(n-i)!} b^i (1-b)^{n-i} \cdot e^{-\lambda kT + (2i-n)\xi} \left\{ e^{-r(n,i)T} K\Phi(-d_2) - S_0\Phi(-d_1) \right\} \quad (4.26)$$

where

$$-d_{1,2} = \frac{\ln(K/S) - \big(r(n,i) \pm \frac{1}{2}\sigma^2\big)\tau}{\sigma\sqrt{\tau}},$$

$r(n,i) \equiv r - \lambda k + \frac{(2i-n)\xi}{\tau}$, and $k \equiv \mathbb{E}U_1 = b\exp(\xi) + (1-b)\exp(-\xi) - 1$.

Theorem 4.3.2. *The price of an American put in the presence of bivariate jumps is given by*

$$V(S,\tau) = p(S,\tau) + rK \int_0^{\tau} e^{-r(\tau-s)} \left[\int_0^{B(s)} \psi(S_s, \tau - s; S)\, dS_s \right] ds \qquad (4.27)$$

where

$$\int_0^{B(s)} \psi(S_s, \tau - s; S) dS_s$$
$$= \sum_{n=0}^{\infty} e^{-\lambda(\tau-s)} \frac{(\lambda(\tau-s))^n}{n!} \sum_{i=0}^{n} \frac{n!}{i!(n-i)!} b^i (1-b)^{n-i}\ \Phi(-d_2) \qquad (4.28)$$

Proof. Similar to the proof of 4.3.1, it is sufficient to show that $\int_0^{B_1} \psi(S_1, \Delta t; S_2)$ is of order Δt or higher. Now, from the following equation:

$$\int_0^{B_1} \psi(S_1, \Delta t; S_2)\, dS_1$$
$$= \sum_{n=0}^{\infty} e^{-\lambda \Delta t} \frac{(\lambda \Delta t)^n}{n!} \sum_{i=0}^{n} \frac{n!}{i!(n-i)!} b^i (1-b)^{n-i}\ \Phi(-d_2) \qquad (4.29)$$

it is not difficult to see that the right side of the equation above is of order Δt or higher. $\square$

4.3.3 Double-exponential density

We assume here that $X = \log(1 + U_1)$ has a double exponential distribution with density

$$f_X(x) = \frac{1}{2\eta} e^{-|x-\kappa|/\eta}, \quad 0 < \eta < 1. \qquad (4.30)$$

Equivalently,

$$X - \kappa = \begin{cases} \xi, & \text{with prob. } 1/2, \\ -\xi, & \text{with prob. } 1/2, \end{cases} \qquad (4.31)$$

where ξ is an exponential random variable with mean η and variance η^2. This is a special case of Bivariate jump distribution in the previous section.

Kou (2000) derives the European put price when the jump size follows double exponential distribution as follows.

$$\begin{aligned}
p(S,\tau) =& Ke^{-r\tau} - S + \sum_{n=1}^{\infty}\sum_{j=1}^{n} e^{-\lambda\tau}\frac{(\lambda\tau)^n}{n!}\frac{2^j}{2^{2n-1}}\begin{pmatrix} 2n-j-1 \\ n-1 \end{pmatrix} \cdot \\
& \Bigg\{ Se^{-\lambda k\tau + n\kappa}\frac{1}{2}\left(\frac{1}{(1-\eta)^j} + \frac{1}{(1+\eta)^j}\right)\Phi(a_1) - e^{-r\tau}K\Phi(a_2) \\
& + \frac{1}{2}e^{-r\tau}e^{-h/\eta}e^{\sigma^2\tau/(2\eta^2)}K\sum_{i=0}^{j-1}\left(\frac{1}{(1-\eta)^{j-1}} - 1\right)\left(\frac{\sigma\sqrt{\tau}}{\eta}\right)^i\frac{1}{\sqrt{2}}Hh_i(c_2) \\
& + \frac{1}{2}e^{-r\tau}e^{h/\eta}e^{\sigma^2\tau/(2\eta^2)}K\sum_{i=0}^{j-1}\left(1 - \frac{1}{(1+\eta)^{j-1}}\right)\left(\frac{\sigma\sqrt{\tau}}{\eta}\right)^i\frac{1}{\sqrt{2}}Hh_i(c_1)\Bigg\} \\
& + e^{-\lambda\tau}\left\{Se^{-\lambda k\tau}\Phi(b_1) - Ke^{-r\tau}\Phi(b_2)\right\},
\end{aligned} \tag{4.32}$$

where

$$\begin{aligned}
a_{1,2} &= \frac{\log(S/K) + (r \pm \frac{\sigma^2}{2} - \lambda k)\tau + n\kappa}{\sigma\sqrt{\tau}}, \\
b_{1,2} &= \frac{\log(S/K) + (r \pm \frac{\sigma^2}{2} - \lambda k)\tau}{\sigma\sqrt{\tau}}, \\
c_{1,2} &= \frac{\sigma\sqrt{\tau}}{\eta} \pm \frac{h}{\sigma\sqrt{\tau}}, \\
h &= \log(K/S) + \lambda k\tau - \left(r - \frac{\sigma^2}{2}\right)\tau - n\kappa, \\
k &= \frac{e^{\kappa}}{1-\eta^2} - 1.
\end{aligned}$$

Remark 4.3.1. Kou (2000) introduces a special function, Hh function, proposed by Laplace in 1774. It is defined by

$$Hh_n(x) = \frac{1}{n!}\int_x^\infty (t-x)^n e^{-t^2/2} dt \quad n = 0, 1, 2, \dots . \tag{4.33}$$

This function has the following properties:

$$\begin{aligned}
Hh_n(x) &= \int_x^\infty Hh_{n-1}(y)dy \\
\frac{d}{dx}Hh_n(x) &= -Hh_{n-1}(x), \quad n = 0, 1, 2, \dots \\
Hh_{-1}(x) &= e^{-x^2/2} = \sqrt{2\pi}\varphi(x) \\
Hh_0(x) &= \sqrt{2\pi}\Phi(-x)
\end{aligned}$$

The Hh function is a useful tool to evaluate the distribution of sum of double exponential and normal random variables.

Theorem 4.3.3. *The American put price, when the jump size follows double exponential distribution, is given by*

$$V(S,\tau) = p(S,\tau) + rK\int_0^\tau e^{-r(\tau-s)}\left[\int_0^{B(s)} \psi(S_s, \tau - s; S)\, dS_s\right] ds \tag{4.34}$$

where

$$\begin{aligned}
\int_0^K \psi(S_\tau, \tau; S)\, dS_\tau = &\sum_{n=1}^\infty \sum_{j=1}^n e^{-\lambda\tau}\frac{(\lambda\tau)^n}{n!}\frac{2^j}{2^{2n-1}}\begin{pmatrix} 2n-j-1 \\ n-1 \end{pmatrix} \cdot \\
&\left\{\Phi(-a_2) + \frac{1}{2}e^{-h/\eta}e^{\sigma^2\tau/(2\eta^2)}\sum_{i=0}^{n-1}\left(\frac{\sigma\sqrt{\tau}}{\eta}\right)^i \frac{1}{\sqrt{2\pi}}Hh_i(c_2)\right. \\
&\left. - \frac{1}{2}e^{h/\eta}e^{\sigma^2\tau/(2\eta^2)}\sum_{i=0}^{n-1}\left(\frac{\sigma\sqrt{\tau}}{\eta}\right)^i \frac{1}{\sqrt{2\pi}}Hh_i(c_1)\right\} + e^{-\lambda\tau}\Phi(-b_2)
\end{aligned} \tag{4.35}$$

Proof. The proof of the first equation is similar to the proof of Theorem 4.1.3. For the proof of the last equation, see Proposition 4 of the paper of Kou (2000), which derives the distribution function of sum of double exponential and the normal random variables. □

Chapter 5

Numerical Implementation and Analysis

5.1 Implementation of previous approaches for American option prices

In this section, we look in detail at two existing models for the valuation of American options proposed by Amin (1993) and Zhang (1997) respectively. Some numerical issues in implementing the methods are also discussed.

5.1.1 Zhang's semi-implicit finite difference method

Zhang's model (1997) applies the variational inequality model for American option pricing in diffusion process introduced by Jaillet, Lamberton, and Lapeyre (1990) to the jump-diffusion case. Zhang's model however is more complicated since the variational inequalities for the jump-diffusion case involve a non local integro-differential operator.

In this section we assume $\mathbb{P}^* = \mathbb{P}$ (i.e., the actual probability measure is an equivalent martingale measure), so that $\mu = r - \lambda \mathbb{E}(U_1)$. Before we introduce the variational inequality, we define the variables used here. We let $v(t, S_t)$ the American option price at time t, and $\psi(x) = f(e^x)$ the payoff function of the option. We also assume that the random variable $Z_j \equiv \log(1 + U_j)$ has a probability density function $g(z)$. Now the value $v(t, x) = u^*(t, \log x)$ of an American option is characterized by the unique solution u^* of the following variational inequality.

$$\begin{gathered} u(T,x) = \psi(x), \\ u(t,x) - \psi(x) \geq 0 \\ \frac{\partial u}{\partial t} + Au + Bu \leq 0 \\ \left[\frac{\partial u}{\partial t} + Au + Bu \right] (u - \psi) = 0 \end{gathered} \tag{5.1}$$

where

$$\begin{aligned} Au &= \frac{\sigma^2}{2}\frac{\partial^2 u}{\partial x^2} - \left(r - \lambda \mathbb{E} U_1 - \frac{\sigma^2}{2} \right) \frac{\partial u}{\partial x} - ru \\ Bu &= \lambda \left(\int_{-\infty}^{\infty} u(t, x+z) g(z) dz - u(t,x) \right) \end{aligned} \tag{5.2}$$

In applying the finite difference method, it is a usual way to define a grid of mesh points $(t,x) = (i\Delta t, j\Delta x)$ where Δt and Δx are mesh parameters which are small and thought of as tending to zero and where i, j are integers such that $i \in \{0, 1, \dots, N\}$ and $j \in \{0, 1, \dots, m+1\}$. By using the finite difference method, Zhang (1997) shows that the discretized system converted from the variational inequality above is given by

$$\begin{gathered} u_0^{i-1} = \psi(m_1 \Delta x), \qquad u_{m+1}^{i-1} = \psi(m_2 \Delta x), \quad 1 \leq i \leq N, \\ \begin{cases} u^N = \phi \\ Mu^{i-1} \geq \hat{q}^i, u^{i-1} \geq \phi \\ (Mu^{i-1} - \hat{q}^i, \phi - u^{i-1}) = 0 \end{cases} \end{gathered} \tag{5.3}$$

with appropriate values of m_1 and m_2 which specify the minimum and maximum stock prices respectively, and with

$$
\begin{aligned}
q_j^i = u_j^i &- (1-\theta)(au_{j-1}^i + bu_j^i + cu_{j+1}^i) \\
&+ \lambda\Delta t(1-\bar{\theta}) \left(\sum_{l+j=0}^{m+1} u_{l+j}^i g_l - u_j^i \right) + \lambda\Delta t \sum_{l+j>m+1,\, l+j<0} \phi_{l+j} g_l. \quad (5.4)
\end{aligned}
$$

where ϕ is a vector with component $\phi_j = \psi((m_1 + j)\Delta x)$ for $1 \le j \le m$, $g_l = g(l\Delta x)$, $\psi(x) = \max(K - e^x, 0)$ for the put options, and M is a $m \times m$ matrix given by:

$$M = \tilde{M} + G \tag{5.5}$$

with

$$
\tilde{M} = \begin{bmatrix}
1+\theta b & \theta c & 0 & \dots & 0 \\
\theta a & 1+\theta b & \theta c & & 0 \\
0 & \theta a & 1+\theta b & & 0 \\
 & \dots & \dots & \dots & \\
 & \dots & \dots & \dots & \\
0 & \dots & \theta a & 1+\theta b & \theta c \\
0 & \dots & 0 & \theta a & 1+\theta b
\end{bmatrix}
$$

where

$$
\begin{aligned}
a &= -\frac{\Delta t \sigma^2}{2(\Delta x)^2} + (r - \lambda \mathbb{E} U_1 - \frac{\sigma^2}{2})\frac{\Delta t}{2\Delta x}, \\
b &= 1 + \frac{\Delta t \sigma^2}{(\Delta x)^2} + r\Delta t, \\
c &= -\frac{\Delta t \sigma^2}{2(\Delta x)^2} - (r - \lambda \mathbb{E} U_1 - \frac{\sigma^2}{2})\frac{\Delta t}{2\Delta x},
\end{aligned}
$$

and

$$G = \bar{\theta}\lambda\Delta t \left(I - \Delta x \begin{bmatrix} g_0 & g_1 & g_2 & \cdots & g_{m-1} \\ g_{-1} & g_0 & g_1 & \cdots & g_{m-2} \\ g_{-2} & g_{-1} & g_0 & \cdots & g_{m-3} \\ & \cdots & \cdots & \cdots & \\ g_{-m+2} & g_{-m+3} & \cdots & \cdots & g_1 \\ g_{-m+1} & g_{-m+2} & \cdots & \cdots & g_0 \end{bmatrix} \right)$$

The vector $\hat{q}$ is defined as follows:

$$\begin{cases} \hat{q}_1 = q_1 - \theta a \psi(m_1 \Delta x) + y_1 \\ \hat{q}_m = q_m - \theta c \psi(m_2 \Delta x) + y_m \\ \hat{q}_j = q_j + y_j, \quad 2 \leq j \leq m-1 \end{cases} \tag{5.6}$$

where the vector $[y_j]$ is given by:

$$y_j = \lambda\bar{\theta}\Delta t \Delta x g_{-j} \psi(m_1 \Delta x) + \lambda\bar{\theta}\Delta t \Delta x g_{m-j+1} \psi(m_2 \Delta x) \quad 1 \leq j \leq m. \tag{5.7}$$

The discretization scheme we consider here is a semi-implicit ($\theta = 1, \bar{\theta} = 0$) in a sense that a fully implicit scheme would lead to full matrices, whereas this scheme involves only tridiagonal matrices. Therefore, the q_j^i is simplified to be:[1]

$$q_j^i = u_j^i + \lambda\Delta t \left(\sum_{l+j=0}^{m+1} u_{l+j}^i g_l - u_j^i \right) + \lambda\Delta t \sum_{l+j>m+1,\, l+j<0} \phi_{l+j} g_l. \tag{5.8}$$

The matrix form of the above formula after changing the dimension ($\{j : 0 \leq j \leq m+1\} \rightarrow \{j : 1 \leq j \leq m+1\}$) for the implementational purpose is given

[1]The notation of q_j^i in Zhang (1997) is wrong, where Δx should be removed.

by:

$$
\begin{bmatrix} g_2 \\ q_3 \\ q_4 \\ \cdot \\ \cdot \\ q_m \end{bmatrix}
=
\begin{bmatrix} u_2 \\ u_3 \\ u_4 \\ \cdot \\ \cdot \\ u_m \end{bmatrix}
+ \lambda \Delta t \left(
\begin{bmatrix}
g_0 & g_1 & g_2 & \cdots & g_{m-2} \\
g_{-1} & g_0 & g_1 & \cdots & g_{m-3} \\
g_{-2} & g_{-1} & g_0 & \cdots & g_{m-4} \\
 & \cdots & \cdots & \cdots & \\
 & \cdots & \cdots & \cdots & \\
g_{-m+2} & g_{-m+3} & \cdots & \cdots & g_0
\end{bmatrix}
\begin{bmatrix} u_2 \\ u_3 \\ u_4 \\ \cdot \\ \cdot \\ u_m \end{bmatrix}
-
\begin{bmatrix} u_2 \\ u_3 \\ u_4 \\ \cdot \\ \cdot \\ u_m \end{bmatrix}
\right)
$$

$$
+ \lambda \Delta t
\begin{bmatrix}
g_{-m} & g_{-m+1} & g_{-m+2} & \cdots & g_{-1} \\
0 & g_{-m} & g_{-m+1} & \cdots & g_{-2} \\
0 & 0 & g_{-m} & \cdots & g_{-3} \\
 & \cdots & \cdots & \cdots & \\
 & \cdots & \cdots & \cdots & \\
0 & 0 & \cdots & g_{-m} & g_{-m+1}
\end{bmatrix}
\begin{bmatrix} \phi_{-m+1} \\ \phi_{-m+2} \\ \phi_{-m+3} \\ \cdot \\ \phi_{-1} \\ \phi_0 \end{bmatrix}
$$

$$
+ \lambda \Delta t
\begin{bmatrix}
g_{m-1} & 0 & 0 & \cdots & 0 \\
g_{m-2} & g_{m-1} & 0 & \cdots & 0 \\
g_{m-3} & g_{m-2} & g_{m-3} & \cdots & \\
 & \cdots & \cdots & \cdots & \\
 & \cdots & \cdots & \cdots & \\
g_1 & g_2 & g_3 & \cdots & g_m
\end{bmatrix}
\begin{bmatrix} \phi_m \\ \phi_{m+1} \\ \phi_{m+2} \\ \cdot \\ \cdot \\ \phi_{2m-1} \end{bmatrix}
\tag{5.9}
$$

where $\phi_j = \psi(m_1 + j\Delta x) = \max\{K - e^{m_1 + j\Delta x}, 0\}$. Note that at the maturity, the value f_j is equivalent to ϕ_{j-1} for $1 \leq j \leq m+1$. When the jump size follows a log-normal distribution, in other words, if $\log(1+U_j)$ follows a normal distribution with mean ζ and variance of δ^2, then $g_l \equiv g(l\Delta x)$ is computed as follows.

$$
g_l = \Phi\Big(\frac{(l+1/2)\Delta x - \zeta}{\delta}\Big) - \Phi\Big(\frac{(l-1/2)\Delta x - \zeta}{\delta}\Big), \tag{5.10}
$$

where $\Phi(x)$ is the standard normal disribution function. We implemented Zhang's discretization scheme introduced in this subsection by using C language and the American option prices under jump-diffusion processes based on this model are shown in the Appendix.

5.1.2 Amin's binomial lattice method

Amin's method (1993) is a discrete-time model based on the Cox, Ross, and Rubinstein (1979, hereafter CRR), and superimposes the jumps on the binomial stock movements. As in the CRR model, Amin's method supposes that the stock price can either move up from S to a new level S^u or down from S to a new level S^d ($S^d < S < S^u$) in a unit period. Unlike the CRR model which permits only a single tick in a unit period, Amin allows the stock price to change by multiple ticks. Thus the diffusion part of a continuous time model is characterized by a single tick, and the jump part by multiple ticks. Amin regards the single tick as a local price change and multiple ticks by a nonlocal price change or a jump. As in the continuous time model, the method incurs an incomplete market in that the jump risk cannot be hedged. Following Merton (1976), Amin assumes that the jump risk is diversifiable and so it is not priced in the market.

Now we will introduce the discrete time model of Amin (1993) in more detail. In this discrete time securities market, tradres occur only on discrete dates indexed by $0, 1, 2, \ldots, T$. The stock price at date i can take on values only in a discrete set specified exogeneously by $S_j(i)$ where $j \in \{-\infty, \ldots, -1, 0, 1, 2, \ldots, \infty\}$. The stock price in state j at date i moves to either state $j+1$ or state $j-1$ at date $i+1$. However when a jump occurs, the stock price moves to potentially any state on the state space grid at the next date. These two price changes are assumed to be mutually exclusive.

Let the price of an option at time i and state j be denoted by $C_j(i)$. Also let Y denote the capital gains return on the stock when a jump occurs and y denote the state induced by this jump at the next date. So if a jump occurs at date i, the stock price at date $i+1$ will become $S_y(i+1) = S(i)Y$. Further, let $\hat{\lambda}$ denote the probability of a jump (under the actual probability measure) in the given time, and E_Y be the expectation operator with respect to distribution of Y.

As in the CRR model, Amin constructs a hedge portfolio with one option, N shares of stock and B dollars of riskless bonds so that the initial value of portfolio can be zero.

$$V(i) = NS(i) + B + C(i) = 0. \tag{5.11}$$

As stated earlier, the hedge portfolio is not guaranteed to be riskless due to an occurrence of a jump. However the jump risk is assumed to be diversifiable as in the Merton's model (1976). Let $V_j(i)$ denote the value of the hedge portfolio at time i and state j. Then, taking the expectation of the hedge portfolio value at date $i+1$ with respect to a jump occurrence and equating it to zero yields

$$0 = \hat{\lambda} E_Y[V_y(i+1)] \; + \; (1-\hat{\lambda}) \; V_{\pm 1}(i+1). \tag{5.12}$$

By selecting the B and S which only hedge the local change of stock price (a single tick), Amin (1993) shows that the option value at date i in terms of its values at date $i+1$ is represented by the following dynamic programming equation:

$$\hat{r}C(i) = \hat{\lambda}E_Y[C_y(i+1)] \ + \ (1-\hat{\lambda})[\hat{q}C_{+1}(i+1) + (1-\hat{q})C_{-1}(i+1)], \quad (5.13)$$

where $\hat{r}$ is a riskless rate of return in every period, and $\hat{q}$ which denotes the probability of a single "uptick" is given by

$$\hat{q} = \frac{(\hat{r} - \hat{\lambda}E_Y[Y])/(1-\hat{\lambda}) - \Delta_{-1}}{\Delta_{+1} - \Delta_{-1}}, \quad (5.14)$$

where $\Delta_k = S_k(i+1)/S(i)$. It is assumed above that $j = 0$ without loss of generality.

Now, the next step of Amin's work (1993) is to define the stock price process in the discrete time model which converges weakly to the continuous time jump-diffusion process proposed by Merton (1976). By defining the drift of the logarithm of the stock price by $\alpha = r - \lambda \mathbb{E}U_1 - \frac{1}{2}\sigma^2$, the new process $X(t)$ under the risk-neutral measure is given by

$$X(t) = \ln\left[\frac{S(t)}{S(0)}\right] = \alpha t + \sigma W(t) + \sum_{j=1}^{N_t} \ln(Y(j)) \quad (5.15)$$

The objective here is to construct a discrete time process which can be used as a suitable approximation to $X(t)$. By partitioning the trading interval $[0,\tau]$ into n subintervals of length $\Delta t = \tau/n$, the stock price in date i and state j is given by

$$S_j(i) = S(0)\exp[\alpha i \Delta t + j\sigma\sqrt{\Delta t}]. \quad (5.16)$$

This state space defined above for the stock prices in the discrete time model corresponds to one first suggested by Jarrow and Rudd (1983).[2]

Amin (1993) also suggests the transition probabilities of X_n conditional on a local price changes and jumps under risk neutral measure as follows:

$$\begin{aligned}\mathrm{Prob}[X_n(t+\Delta t) - X_n(t) = \alpha\Delta t + \sigma\sqrt{\Delta t}] &= q_n(1-\lambda\Delta t)\\ \mathrm{Prob}[X_n(t+\Delta t) - X_n(t) = \alpha\Delta t - \sigma\sqrt{\Delta t}] &= (1-q_n)(1-\lambda\Delta t)\\ \mathrm{Prob}[X_n(t+\Delta t) - X_n(t) = \alpha\Delta t + l\sigma\sqrt{\Delta t}; l \neq \pm 1] &= \lambda\Delta t h(l)\end{aligned} \tag{5.19}$$

for all $t \in [0, \Delta t, 2\Delta t, \ldots, \tau - \Delta t]$, where q_n is shown to be the following:

$$q_n = 1/2 + O(\Delta t) \tag{5.20}$$

Let the cumulative density function of $\ln Y$ under the risk neutral measure be

[2]In constructing a binomial tree, Jarrow and Rudd (1983) use different approaches to selecting the parameter values. In the binomial model described below,

$$S_{t+1} = \begin{cases} S_t e^u & \text{with probability } q \\ S_t e^d & \text{with probability } 1-q \end{cases} \tag{5.17}$$

where u, d, q are constants which satisfy $u > r\Delta t > d$ and $0 < q < 1$, they use the following values to obtain the approximation to the Black-Scholes formula:

$$\begin{aligned} q =& 1/2 \\ u =& (r - \sigma^2/2)\Delta t + \sigma\sqrt{\Delta t} \\ d =& (r - \sigma^2/2)\Delta t - \sigma\sqrt{\Delta t} \end{aligned} \tag{5.18}$$

Jarrow and Rudd (1983) mention that their choice of the parameters $\{q, u, d\}$ insures that both the binomial and the lognormal process have the same first two moments as $\Delta t \to 0$, while the Cox, Ross, and Rubinstein (1979) choice of parameters ensures equality of the first moment, but the variances of the two processes are equal only in the limit. (For the proof, see page 184-186 of Jarrow and Rudd (1983).)

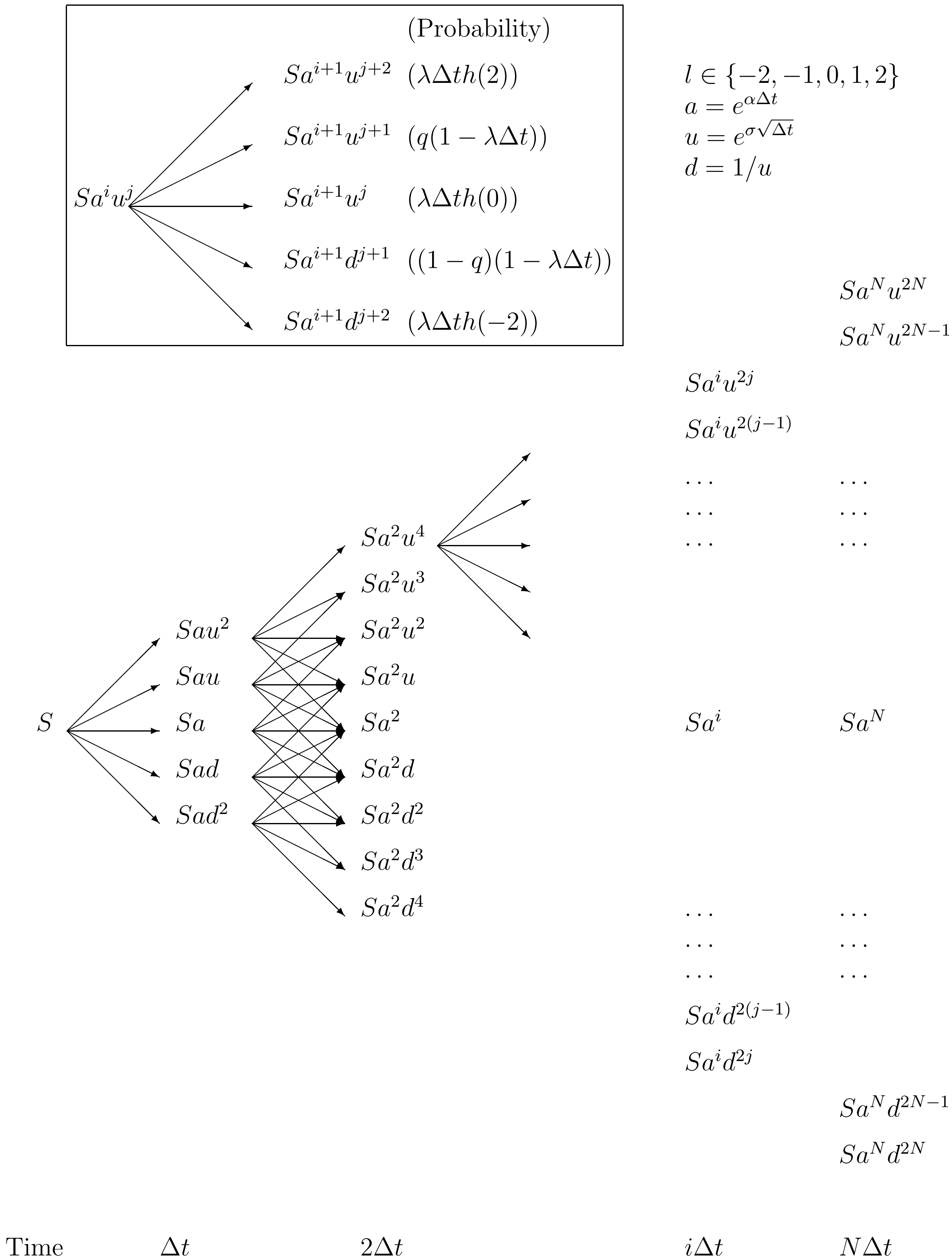


Figure 5.1: Dynamics of stock price in Amin's method

given by $\mathcal{N}(x)$ for $x \in \mathbb{R}$. Then, $h(l)$ can be denoted by:

$$\begin{aligned}
h(l) &= \mathcal{N}\Big(\alpha\Delta t + (l+1/2)\sigma\sqrt{\Delta t}\Big) - \mathcal{N}\Big(\alpha\Delta t + (l-1/2)\sigma\sqrt{\Delta t}\Big) \\
&\qquad \text{if } l \notin \{-1,0,1\} \\
h(0) &= \mathcal{N}\Big(\alpha\Delta t + (1+1/2)\sigma\sqrt{\Delta t}\Big) - \mathcal{N}\Big(\alpha\Delta t - (1+1/2)\sigma\sqrt{\Delta t}\Big) \\
h(\pm 1) &= 0
\end{aligned}$$

It is assumed here that multiple jumps at any discrete date cannot occur. Given the discrete approximation, one can compute option prices by dynamic programming using a backward recursion on the state space described. Amin (1993) suggests that the possible values of l in Equation (5.19) can be selected by truncating the jump distribution outside the region $[\alpha\Delta t - L\sigma\sqrt{\Delta t}, \alpha\Delta t + L\sigma\sqrt{\Delta t}]$, where "$L$" is the smallest nonnegative integer such that this interval contains $[-3\delta, 3\delta]$, where δ is the variance of $\ln Y$. For instance, when the numbers of time steps(N) are chosen as $N = 50, 100, 200$, the required values of L are given by 68, 96, and 135 respectively for the numerical example in Table B.2. Figure 5.1.2 illustrates the state space for the discrete model for the simple case of $L = 2$.

5.2 Implementation of new approaches

In this section, we introduce several numerical methods to compute critical stock prices in a more efficient way. All the methods we mention here were previously proposed to obtain American option prices in the diffusion case. We however apply the methods to the jump-diffusion case for the first time in the literature.

5.2.1 Extended integral equation method

We need to know the critical stock prices ahead before computing the American option prices. When the current stock price is equal to the critical stock price, the "value matching" condition leads to the following integral equation.

$$K - B(\tau) = p(B(\tau), \tau) + rK \int_0^{\tau} e^{-r(\tau - s)} \left[\int_0^{B(s)} \psi(S_s, \tau - s; B(\tau))\, dS_s \right] ds \tag{5.21}$$

As pointed out by Kim (1990), the integral equation is Volterra type and of the second kind, and so it can be solved by numerical method. We divide the time to maturity τ into N subintervals:

$$t_i = i\Delta t, \qquad i = 0, 1, \ldots, N, \qquad \Delta t \equiv \frac{\tau}{N} \tag{5.22}$$

We define the integrand of the outer integral part of the Equation (5.21) as the following function:

$$T(t_i, s) = e^{-r(t_i - s)} \int_0^{B(s)} \psi\Big(S_s, t_i - s; B(t_i)\Big)\, dS_s.$$

The trapezoidal rule method yields:

$$\int_0^{t_i} T(t_i, s) ds = \Delta t \left(\frac{1}{2} T_{i0} + \sum_{j=1}^{i-1} T_{ij} + \frac{1}{2} T_{ii} \right) \tag{5.23}$$

where

$$\begin{aligned} T_{ij} &= e^{-r(t_i - t_j)} \int_0^{B_j} \psi\Big(S_j, t_i - t_j; B_i\Big)\, dS_j \\ &= e^{-r(i-j)\Delta t} \int_0^{B_j} \psi\Big(S_j, (i-j)\Delta t; B_i\Big)\, dS_j \end{aligned} \tag{5.24}$$

Thus the trapezoidal method for the entire integral equation yields:

$$\begin{aligned} K - B_0 &= 0 \\ K - B_i &= p(B_i, t_i) + rK\Delta t \left(\frac{1}{2} T_{i0} + \sum_{j=1}^{i-1} T_{ij} + \frac{1}{2} T_{ii} \right) \end{aligned} \tag{5.25}$$

for $\leq i \leq N$. More explicitly, we can represent the previous equations as follows:

$$\begin{aligned} K - B_0 &= 0. \\ K - B_1 &= p(B_1, \Delta t) + \frac{1}{2} e^{-r\Delta t} rK\Delta t \int_0^{B_0} \psi(S_0, \Delta t; B_1)\, dS_0 \\ K - B_2 &= p(B_2, 2\Delta t) + \frac{1}{2} e^{-2r\Delta t} rK\Delta t \int_0^{B_0} \psi(S_0, 2\Delta t; B_2)\, dS_0 \\ &\quad + e^{-r\Delta t} rK\Delta t \int_0^{B_1} \psi(S_1, \Delta t; B_2)\, dS_1 \\ &\vdots \\ &\vdots \\ K - B_N &= p(B_N, N\Delta t) + \frac{1}{2} e^{-rN\Delta t} rK\Delta t \int_0^{B_0} \psi(S_0, N\Delta t; B_N)\, dS_0 \\ &\quad + rK\Delta t \sum_{i=1}^{N-1} \left[e^{-(N-i)r\Delta t} \int_0^{B_i} \psi(S_i, (N-i)\Delta t; B_N)\, dS_i \right] \end{aligned} \tag{5.26}$$

In the second equation of (5.26), B_1 is the only unknown value given that $B_0 = K$ from the first equation. Since this is a nonlinear equation, one can solve it using Newton-Raphson method. In a similar way, the set of critical stock prices, $\{B_i\}_{0 \leq i \leq N}$, can be obtained recursively. Although every right-hand side of the equations is formulated as the infinite summation of Poisson distribution, we restrict the number of jumps appropriately and can handle the equations without difficulty.

5.2.2 Modified MacMillan-Zhang's analytical method

Another approach to obtaining the critical stock prices is based on the the analytical option pricing model by Zhang (1995), which is a jump-diffusion version of MacMillan's analytical method (1986) for the pure diffusion case. The critical stock prices with time to maturity of $i\Delta t$ for $i \in \{1, 2, \ldots, N\}$ in the jump-diffusion case can be obtained by using the two steps.

- Step I: Compute negative value of η satisfying $\phi(\eta) = 0$, where:

$$\phi(\alpha) = \frac{\sigma^2}{2}\alpha^2 + (\mu - \frac{\sigma^2}{2})\alpha - (r + \lambda + \frac{1}{i\Delta t}) + \lambda e^{\frac{\alpha^2\delta^2}{2} + \alpha m} \tag{5.27}$$

  for $i = 1, 2, \ldots, N$.

- Step II: Compute the critical stock price x^* satisfying $f(x) = x$, where

$$f(x) = |\eta| \frac{K - p(x, i\Delta t)}{p^{'}(x, i\Delta t) + 1 + |\eta|} \tag{5.28}$$

  where $p(x, i\Delta t)$ is a European option price, and $p^{'}(x, i\Delta t) = \frac{\partial p}{\partial x}$.

Remark 5.2.1. The jump-diffusion version of analytical method of Zhang (1995) is composed of four steps to yield the value of American option. The above two steps are exactly identical to the first two steps of the Zhang's original model. However as shown in the Table B.2, one can see that the analytical method introduced in this subsection is more accurate than Zhang's.

5.2.3 Extrapolation method

In computing American option prices, we estimate the early exercise boundary first either by integral equation method or by analytical method. In order to approximate the early exercise boundary to the true value as possible as we can, we need many numbers of time steps. However this requires lots of CPU times. In this section, we suggest a method to allow more accuracy with less computing times. Geske and Johnson (1984) proposed a Richardson scheme to extrapolate the American option price P_0 as follows (for example, three-point Richardson extrapolation scheme):

$$\hat{P}_0 = (P_1 - 8P_2 + 9P_3)/2. \tag{5.29}$$

where P_i, $i = 1, 2, 3$, is the price of an i-times exercisable option, and $\hat{P}_0$ denotes an estimate of P_0. The numbers of time steps needed to obtain the three prices (P_1, P_2, and P_3) are $0, 1$, and 2, respectively. As we increase the number of points, we can obtain more accurate value.

5.2.4 Extended approximating method by a multipiece exponential function

We have already shown that the price of American put with time to maturity of T and current stock price of S is given by

$$V(S,T) = p(S,T) + rK \int_0^T e^{-r(T-t)} \left[\int_0^{B_t} \psi(S_t, T-t; S)\, dS_t \right] dt,$$

where B_t and S_t are the stock price and critical stock price with time to maturity of t. If we transform B_t and S_t by replacing t with $T - t$, then we obtain the following formula:

$$V(S,T) = p(S,T) + rK\int_0^T e^{-rt}\left[\int_0^{B_t}\psi(S_t,t;S)\,dS_t\right]dt,$$

which is the same form that Ju uses in Equation (1) of his paper (1998). According to the value match condition, the critical stock price B_t solves the following integral equation:

$$K - B_t = p(B_t, T-t) + rK\int_t^T e^{-r(s-t)}\left[\int_0^{B_s}\psi(S_s, s-t; B_t)\,dS_s\right]ds, \tag{5.30}$$

We assume that the critical stock price, B_t, is an exponential function $B\exp(bt)$ for the interval $[t_1, t_2]$, and define the the integral:

$$I(t_1, t_2, S, B, b) = \int_{t_1}^{t_2} e^{-rt}\left[\int_0^{Be^{bt}}\psi(S_t,t;S)\,dS_t\right]dt \tag{5.31}$$

Note that the above integral can be converted analytically as the sum of normal distribution functions if there is no jump. However in the presence of jumps, the integral part can be evaluated by using trapezoidal rule.

Now we define P_1, P_2, P_3, etc., as the approximate American put values corresponding to approximating the early exercise boundary as a one-piece exponential function, a two-piece exponential function, and a three-piece exponential function, etc., then the $P's$ are given by:

$$\begin{aligned}
P_1 &= p(S,T) + rKI(0,T,S,B_{11},b_{11})\\
P_2 &= p(S,T) + rKI(0,T/2,S,B_{22},b_{22}) + KI(T/2,T,S,B_{21},b_{21})\\
P_3 &= p(S,T) + rKI(0,T/3,S,B_{33},b_{33}) + KI(T/3,2T/3,S,B_{32},b_{32})\\
&\qquad + KI(2T/3,T,S,B_{31},b_{31})
\end{aligned} \tag{5.32}$$

To determine $B's$ and $b's$, we apply the value-match, and high-contact conditions, and solve them using Newton-Raphson method. We also use a three-point Richardson scheme explained in the previous section. The three point Richardson scheme gives the approximate American put price as follows:

$$\hat{P_A} = 4.5P_3 - 4P_2 + 0.5P_1 \tag{5.33}$$

First, consider the value P_1, which uses only one-piece exponential function. Applying value-match and high-contact condition at $t = 0$ would yield the following equations:

$$\begin{aligned} K - B_{11} &= p(B_{11}, T) + rKI(0, T, B_{11}, B_{11}, b_{11}) \\ -1 &= -N(-d_1(B_{11}, K, T)) + rKI_S(0, T, B_{11}, B_{11}, b_{11}) \end{aligned} \tag{5.34}$$

Next, consider the value P_2, which uses two-piece exponential function. In the same manner, the value-match and high-contact conditions at $t = T/2$ yield the equations:

$$\begin{aligned} K - B_{21}e^{b_{21}T/2} &= p(B_{21}e^{b_{21}T/2}, T/2) \\ &\quad + rKI(0, T/2, B_{21}e^{b_{21}T/2}, B_{21}e^{b_{21}T/2}, b_{21}) \\ -1 &= -N(-d_1(B_{21}e^{b_{21}T/2}, K, T/2)) \\ &\quad rKI_S(0, T/2, B_{21}e^{b_{21}T/2}, B_{21}e^{b_{21}T/2}, b_{21}) \end{aligned} \tag{5.35}$$

The remaining part of exponential function can be obtained as follows by evaluating at $t = 0$:

$$
\begin{aligned}
K - B_{22} &= p(B_{22}, T) \\
&\quad + rKI(0, T/2, B_{22}, B_{22}, b_{22}) + rKI(T/2, T, B_{22}, B_{21}, b_{21}) \\
-1 &= -N(-d_1(B_{22}, K, T)) \\
&\quad + rKI_S(0, T/2, B_{22}, B_{22}, b_{22}) + rKI_S(T/2, T, B_{22}, B_{21}, b_{21})
\end{aligned}
\tag{5.36}
$$

Similarly, the same technique applied to the value of P_3 gives

(i) at $t = 2T/3$:

$$
\begin{aligned}
K - B_{31}e^{b_{31}2T/3} &= p(B_{31}e^{b_{31}2T/3}, 2T/3) \\
&\quad + rKI(0, T/3, B_{31}e^{b_{31}2T/3}, B_{31}e^{b_{31}2T/3}, b_{31}) \\
-1 &= -N(-d_1(B_{31}e^{b_{31}2T/3}, K, 2T/3)) \\
&\quad + rKI_S(0, T/3, T, B_{31}e^{b_{31}2T/3}, B_{31}e^{b_{31}2T/3}, b_{31})
\end{aligned}
\tag{5.37}
$$

(ii) at $t = T/3$:

$$
\begin{aligned}
K - B_{32}e^{b_{32}T/3} &= p(B_{32}e^{b_{32}T/3}, 2T/3) \\
&\quad + rKI(0, T/3, B_{32}e^{b_{32}T/3}, B_{32}e^{b_{32}T/3}, b_{32}) \\
&\quad + rKI(T/3, 2T/3, B_{32}e^{b_{32}T/3}, B_{31}e^{b_{31}T/3}, b_{31}) \\
-1 &= -N(-d_1(B_{32}e^{b_{32}T/3}, K, 2T/3)) \\
&\quad + rKI_S(0, T/3, B_{32}e^{b_{32}T/3}, B_{32}e^{b_{32}T/3}, b_{32}) \\
&\quad + rKI_S(T/3, 2T/3, B_{32}e^{b_{32}T/3}, B_{31}e^{b_{31}T/3}, b_{31})
\end{aligned}
\tag{5.38}
$$

(iii) at $t = 0$:

$$
\begin{aligned}
K - B_{33} &= p(B_{33}, T) \\
&\qquad + rKI(0, T/3, B_{33}, B_{33}, b_{33}) + rKI(T/3, 2T/3, B_{33}, B_{32}, b_{32}) \\
&\qquad + rKI(2T/3, T, B_{33}, B_{31}, b_{31}) \\
-1 &= -N(-d_1(B_{33}, K, T)) \\
&\qquad + rKI_S(0, T/3, B_{33}, B_{33}, b_{33}) + rKI_S(T/3, 2T/3, B_{33}, B_{32}, b_{32}) \\
&\qquad + rKI_S(2T/3, T, B_{33}, B_{31}, b_{31})
\end{aligned}
\tag{5.39}
$$

For each set of equation, one can use two-dimensional Newton-Raphson method to compute the unknown parameters ($B's$ and $b's$). Once the unknown parameters are found, the prices of American puts can be obtained from Equation (5.32).

5.2.5 Implementation for the general case

In the previous scetion, we assume that $\eta = 0$. In other words, the actual probability measure is assumed to be an equivalent martingagle measure. If $\eta \neq 0$, then we however need to adjust the values of the parameters under the new minimal martingale measure, since we only know the parameter values under the actual measure. First assume here that $\log(1 + U_1)$ follows a normal distribution with mean m and variance δ^2 under the actual measure $\mathbb{P}$ given the condition that $\log(1 + U_1)$ follows a normal distribution with mean $\hat{m}$ and variance $\hat{\delta}^2$ under the minimal martingale measure $\mathbb{Q}$. Then the first and second moment ($n = 1, 2$) of $\log(1 + U_1)$ under $\mathbb{Q}$ are given by:

$$
\begin{aligned}
\mathbb{E}^{\mathbb{Q}}[(\log(1+U_1))^n] &= \int_{\mathbb{R}} (\log(1+x))^n \, d\mathbb{Q}_{U_1}(x) \\
&= \int_{\mathbb{R}} (\log(1+x))^n \, \frac{1-\eta x}{1-\eta \mathbb{E} U_1} d\mathbb{P}_{U_1}(x) \\
&= \frac{\mathbb{E}[(\log(1+U_1))^n] - \eta \mathbb{E}[(\log(1+U_1))^n U_1]}{1-\eta \mathbb{E} U_1}
\end{aligned}
\tag{5.40}
$$

Next, as shown in the previous section, k^* and λ^* are given by

$$
\begin{aligned}
k^* &= \mathbb{E}^{\mathbb{Q}} U_1 = \frac{\mathbb{E} U_1 - \eta \mathbb{E} U_1^2}{1-\eta \mathbb{E} U_1} \\
\lambda^* &= \lambda(1-\eta \mathbb{E} U_1),
\end{aligned}
\tag{5.41}
$$

where $\mathbb{E} U_1^2 = (\mathbb{E} U_1 + 1)^2 e^{\delta^2} - 2\mathbb{E} U_1 - 1$.

Remark 5.2.2. For example, when $\mathbb{E} U_1 = 0$, which imply $m = -\delta^2/2$, the mean and variance of $\log(1 + U_1)$ under $\mathbb{Q}$ are given by

$$
\begin{aligned}
\mathbb{E}^{\mathbb{Q}}[\log(1+U_1)] &= m(1+2\eta) \\
\mathbb{V}^{\mathbb{Q}}[\log(1+U_1)] &= \delta^2 - 4m^2\eta(1+\eta)
\end{aligned}
$$

5.3 Numerical comparisons

In this section we present three families of numerical comparisons. The parameter values for two examples are taken from the numerical examples in the papers of Amin(1993), and Zhang(1997). In computing the prices, we use a cluster of computer systems known as UNIX Timesharing Services (UTS) in University of Texas at Austin which consists of two Digital Equipment

Corporation AlphaServer 2100 4/275s. Each 2100 machine has dual 275MHz of CPU, 512 MB of memory, 64-bit architecture, and 42GB (RAID 5) of home disk space.

In most numerical examples except Tables B.4 and B.5 , we consider only the simple case with having $k = 0$ so that the total variance is given by: $\sigma^2_{total} = \sigma^2 + \lambda(e^{\delta^2} - 1)$. The following set of parameter values are used in the numerical examples.

Case I:

Initial stock price (S) = (\$80, 90, 100, 110, 120), exercise price (X) = \$100, annual risk-free interest rate (r) = 6%, maturity (τ) = (0.25, 1.0) year, annual variance of diffusion component(σ^2) = 0.09, variance of stock price return due to each jump occurrence(δ^2) = 0.0225, jump intensity(λ) = 1.0, mean of relative jump size(k) = 0

Case II: Amin's example (p.1853, 1993)

Initial stock price (S)= \$40, exercise price (X) = (30, 35, 40, 45, 50), annual risk-free interest rate (r) = 8%, time to maturity (τ) = (0.25, 1.0) year, annual variance of diffusion component(σ^2) = 0.05, variance of stock price return due to each jump occurrence(δ^2) = 0.05, jump intensity(λ) = 5.0, mean of relative jump size(k) = 0.

Case III: Zhang's example (p.688, 1997)

Initial stock price (S) = (\$40, 45, 50, 55), exercise price (X) = \$45, annual risk-free interest rate (r) = 9%, maturity (τ) = (0.25, 0.5, 1.0) year, annual variance of diffusion component(σ^2) = 0.004, variance of stock price return due to each jump occurrence(δ^2) = 0.039220713, jump intensity(λ) = 0.9, mean of relative jump size(k) = 0

5.4 Analysis of algorithms

In this section, we analyze the three aspects of each algorithm:

(i) The complexity (running time) of the algorithms.

(ii) The precision of the algorithms.

(iii) The rate of convergence with respect to the number of time steps (N).

5.4.1 Comparing complexity of algorithms

The complexity of our new algorithm based on the integral equation is $O(N^2)$, where N is the number of time steps discretized. The complexity of the Amin's algorithm is $O(N^2L^2)$, where N is the number of time steps, and L is the number of jump states for each state. If $N = L$, then the complexity will be $O(N^4)$. If we use the truncation method described in the paper, then the complexity decrease to $O(N^2L)$. However the accuracy of the option prices obtained from this scheme is problematic. The complexity of the Zhang's method is $O(NM^2)$, where N is the number of time steps, and M is the number of stock price meshed. In summarizing, the extended integral equation method improves the computational efficiency over the other methods. Especially the jump process does not affect the complexity of the new algorithm, even if the running time is much greater than the integral equation method for the Black-Scholes model. Table B.6 shows the required CPU times to compute option prices with respect to N, L and M, and one can see that they vary corresponding to the numerical complexities of the algorithms.

5.4.2 Comparing precision

Once we implement a method to compute the exact price of option with jumps, we can compare the accuracy. Unfortunately the exact price is not known. The American option prices are known only when there is no jump processes. Since $\sigma_{total}^2 = \sigma^2 + \lambda \mathbb{E} U_1^2$, the total volatility is the same as the volatility of diffusion component when the jump intensity is zero or the jump volatility is zero. In such circumstances, the American option price under pure diffusion processes is equivalent to that under jump-diffusion processes. Thus we take examples from two extreme cases: (1) the jump intensity is very small, and (2) the jump volatility is very small. In both cases it is reasonable to regard the binomial price with more than 10,000 numbers of time steps as the true value of the option prices since the jump effect is very negligible.

We define RMSE (Relative root-mean-square error) as

$$\sqrt{\frac{1}{n} \sum \left(\frac{p - \hat{p}}{p} \right)^2}$$

where n is the number of samples, p is the true price computed by binomial model with 10,000 time steps, and $\hat{p}$ is the price obtained by one of three methods. The number of samples is 10, and the numbers of time steps and stock price meshed for three methods are as follows:

- Extended integral equation method: $N = 10, 20, 40, 50, 80, 100$
- Amin's method: $N = 50, 80, 100, 150, 200, 250$, and $M = N/2$
- Zhang's method: $N = M = 40, 60, 80, 100, 150, 200$

Figure B.1 to B.4 illustrate log function of average computing time in x-axis and RMSE in y-axis. They show that the RMSE decreases as the number of time steps decreases or the required computing time is small in most cases. The only exception is the graph of Amin's method in Figure B.1, which is due to the fluctuation property of the prices with respect to the number of time steps. The results clearly reveal that the RMSE of the extended integral equation method, given a specified computing time, is lower than that of the other methods in most cases. Thus we conclude that the extended integral equation method yields more precise option prices than the other methods under a given computing time.

5.4.3 Comparing rate of convergence

Let p_N denote the numerically calculated price of option when the number of time steps is N. The sequence of prices (p_N) converges with order $\rho > 0$ if there exists $k > 0$ such that for all $N \in \mathbb{N}$; $e_N \leq k/N^\rho$, where $e_N \equiv |p_\infty - p_N|$.

If we take log function to both sides, we obtain the following:

$$\ln e_N \leq \ln(k/N^\rho) = \ln k - \rho \ln N \tag{5.42}$$

This equation implies that ρ is the slope of the line generated by $\ln e_N$ and $\ln N$. Figure B.5 to Figure B.36 reveal that our extended integral equation method shows the better performance of convergence to the true price than the other two methods in that the magnitude of errors of prices obtained using the new method is smaller than that of prices obtained using the other methods.

5.5 Properties of prices

(i) Table B.1 to Table B.3 provide American put prices computed in three different ways for several numbers of time steps (N), number of jump states (L), and the number of stock price meshed (M). In most cases, the difference of prices computed using three methods is less than one cent except two cases (with N=100 for the extended integral equation method, $N = L$=200 for the Amin's method, and $N = M = 200$ for the Zhang's method, respectively). When the maturity is one year in Case I, the prices computed using the extended integral equation method is around two cents greater than those computed using Amin's and Zhang's method. Also for the short maturity in Case III, the price computed using Amin's method is more than 40 cents different from the others computed using the new method and Zhang's (see also Figure B.24). In that example, the convergence rate of Amin's method is slower than the other methods, and we need more numbers of time steps (i.e., $N > 200$) to obtain appropriate prices.

(ii) Table B.1 to Table B.3 also contain the MacMillan-Zhang's analytical method, and the modified MacMillan-Zhang's method, three-point Richardson extrapolation method and Ju's approximation method using three-piece exponential function. The three new analytical methods show less RMSE than the previous MacMillan-Zhang's method in three cases, which implies better performance in numerical computation.

(iii) Table B.4 provides the option prices when $k(\equiv \mathbb{E}U_1)$ varies in the range of -0.9 and 0.9. The results show that the price increases as the absolute value of k becomes larger, which is not strange since larger absolute value of k causes larger value of jump volatility. However the effect is not symmetric because negative value of k implies more chance of negative jumps and thus the put option is more likely to be exercised than in the positive k. Thus we can conclude that both volatility and jump effect determine the movement of option prices.

In order to look at the net effect of jumps, we fix the total volatility and change only the level of k, as shown in Table B.5. When k increases, the option prices obtained using our extended integral equation decrease, which is a natural result because the put option has a small value when there is a higher tendency of positive jumps. However the prices using Amin's method seem to converge very slowly when the value of k is less than around -0.8. Even worse, Zhang's method shows totally different results in the same range of k. We will explore those weird behaviors of Amin and Zhang's methods in the future research.

(iv) Table B.6 provides American put prices by varying the ratio of the variance of diffusion component to the total variance ($\sigma^2/\sigma^2_{total}$) We let the total variance fixed but vary the jump intensity from $\lambda = 5$ to $\lambda = 0.0001$. For each level of jump intensity, we adjust the variance of diffusion component so that the total variance can be constant. We find that as we decrease the intensity of jumps gradually to zero or the ratio of the variance of jump component to that of total variance ($\sigma^2_D/\sigma^2_{total}$) to one, the prices appear to converge to the price of the Black-Scholes model.

The numerical examples here do not support Zhang (1997)'s argument that an option price in the presence of jumps is greater than that in the Black-Scholes model when the option is strongly out of the money. The argument is true only when the jump intensity is greater than one. When the jump intensity is less than one, the prices increase as the jump intensity decreases in the range between 0 and 1. When the option is in the money or at the money, however the argument that an option price in the presence of jumps is smaller than that in the Black-Scholes model holds also in these examples.

(v) When η varies between zero and one, the option prices are shown in Table B.11 to Table B.13. The options with greater η appear to have larger prices. It is very natural to obtain such results since greater value of η implies greater expected rate of return over risk-free rate.

(vi) Figure B.5 to Figure B.28 depict American put prices with respect to number of time steps in three different methods. The Amin's price and the finite difference method price in some cases show an oscillatory convergence. Meanwhile the prices obtained using our extended integral equation method in Case I shows a smoother convergence. For the oscillatory convergence one can apply the average method as pointed out by Broadie and Detemple (1996), where $n-$ and $(n+1)-$ steps prices are averaged, which we leave as one of our future research. Also for the smoother convergence one can apply the Richardson extrapolation method as conducted in this thesis to obtain the more closer value to the true value without the expense of greater running times.

(vii) In the multinomial lattice model and the multi-dimensional finite difference method, their computational effort grows exponentially with respect to the number of state variables. The running times of the lattice model and the finite difference method are $O(N^2 2^K)$ and $O(NM2^K)$ respectively, where N is the number of time steps discretized, M is the number of meshed stock prices and K is the number of underlying assets.

The integral method requires to compute beforehand the optimal exercise boundary of each underlying asset with respect to every discrete time steps up to the exercising dates recursively, and so the computational work increases with rate of K^2. Thus the running time of the integral method for the multiple assets is $O(N^2K^2)$. As pointed out by Broadie and Glasserman (1997a,b), the convergence rate of Monte Carlo simulation method is typically independent of the problem dimension and so the running time is given by $O(N^2N_1 \max(N_2, N_3)K)$, where N_1 is the number of outer simulation trials, N_2 and N_3 are the numbers of inner simulation trials. Even in the case that requires much computational work under the jump-diffusion process, the numerical efficiency of the simulation method will be less deteriorated than that of the other approaches.

Chapter 6

Conclusion

6.1 Concluding remarks

In this dissertation we study the numerical valuation of American and European option prices under jump-diffusion processes. Due to the jump part, the market is incomplete and so it is impossible to construct a hedging portfolio with stocks and riskless assets. Contrary to the case of a complete market in which only one equivalent martingale measure exists, there are infinite numbers of equivalent martingale measures in an incomplete market. Our research here is focusing on the well known notion of risk minimizing strategy and its associated minimal martingale measure.

Based on the risk minimizing strategy introduced in the works of Föllmer, Schweizer, and Sondermann (1986, 1990, 1991), we characterize the dynamics of a risky asset and valuation formulas for option prices under jump-diffusion processes. In particular, we obtain an analytical formula for a European option price. But the main contribution of this dissertation is to extend Kim (1990)'s early exercise premium representation based on a decomposition method in order to calculate an American option price under jump-diffusion processes as a summation of a European option price and early exercise premiums.

We derive the early exercise premium representation under jump-diffusion processes with various distributions of jump size - lognormal, jump-to-ruin, bivariate and double exponential distribution. In calculating an optimal boundary, we modify and extend numerical methods previously used in the pure diffusion processes - integral equation method of Kim (1990), and the approximation scheme by multipiece exponential functions of Ju (1998). Also we apply Richardson extrapolation scheme of Geske and Johnson (1984) and modify MacMillan-Zhang's analytical method (1995) to calculate American option prices in a faster way.

We also implement two previous models for pricing under jump diffusions: a binomial lattice model of Amin (1993) and a semi-implicit finite difference method of Zhang (1997) and compare them with our extended integral equation method. The difference of the American option prices computed using the three different methods is less than one cent in most of cases shown in the numerical examples. However we find that the numerical performance of our new method is more efficient than the two existing methods in that the new one shows lower Relative root-mean-square error under the same computing time, possesses a lower degree of algorithmic complexity, and converges faster than the two existing methods.

6.2 Future research

In addition to the risk-minimizing strategy applied in this thesis to the jump-diffusion model, utilizing variance minimizing strategy or expected utility maximizing scheme is also an alternative way for the valuation of options in an incomplete market, and generalizing our numerical procedure to these other strategies is a natural future research activity. Another part of main future works is to extend the numerical procedures developed here to the valuation of American options on multiple underlying assets in a jump-diffusion model. Since the valuation problem in a high dimension has not been fully explored yet in the pure diffusion model, the priority of our research will be given to that problem first. Our future research will focus on the properties of the optimal boundary of a multiple option and the numerical issues on the computation of the optimal boundary and option prices based on the integral equation method.

Appendices

Appendix A

Proofs

A.1 Proof of Equation (3.9)

Proof. Under the assumption of Merton (1976) that the jump risk is nonsystematic, one can see $\mu = r - \lambda k$ from the partial differential equation (3.6). Thus we let $\mathbb{P}$ denote the equivalent martingale measure correspondig to Merton's assumption.

Let $p(S_0, T)$ be the value of European put with jump-diffusion process, and $\mathbb{E}$ be the expectation under the measure $\mathbb{P}$. Then,

$$\begin{aligned} p(S_0, T) =& \mathbb{E}[e^{-rT}(K - S_T)^+] \\ &= \mathbb{E}[e^{-rT} K I_D] - \mathbb{E}[e^{-rT} S_T I_D] \\ &= A - B, \end{aligned} \tag{A.1}$$

where $I_D = I_{\{S_T<K\}}$ is an indicator function. Now,

$$\begin{aligned}
A &= \mathbb{E}[e^{-rT}KI_D] \\
&= e^{-rT}K\mathbb{P}[S_T < K] \\
&= \sum_{n=0}^{\infty} \frac{e^{-\lambda T}(\lambda T)^n}{n!} e^{-rT}K \cdot \\
&\quad \cdot \mathbb{P}\Big[S_0 \exp(\sigma W_T + (r - \lambda k - \frac{1}{2}\sigma^2)T + \sum_{j=1}^{n} \ln(1+U_j)) < K\Big] \\
&= \sum_{n=0}^{\infty} \frac{e^{-\lambda T}(\lambda T)^n}{n!} e^{-rT}K \\
&\quad \cdot \mathbb{P}\Big[\sigma W_T + \sum_{j=1}^{n} \ln(1+U_j) < \ln(K/S_0) - (r - \lambda k - \frac{1}{2}\sigma^2)T\Big]
\end{aligned}$$

Since W_T follows a standard Gaussian law $N(0,T)$, and $\ln(1+U_j)$ follows $N(m,\delta^2)$ under measure $\mathbb{P}$, one can see that

$$\begin{aligned}
A &= \sum_{n=0}^{\infty} \frac{e^{-\lambda T}(\lambda T)^n}{n!} e^{-rT}K\mathbb{P}\Big[\xi < \frac{\ln(K/S_0) - (r - \lambda k - \frac{1}{2}\sigma^2)T - nm}{\sqrt{\sigma^2 T + n\delta^2}}\Big] \\
&= \sum_{n=0}^{\infty} \frac{e^{-\lambda T}(\lambda T)^n}{n!} e^{-rT}K\mathbb{P}\Big[\xi < \frac{\ln(K/S_0) - (r - \lambda k - \frac{1}{2}\sigma^2 + \frac{nm}{T})T}{\sqrt{\sigma^2 T + n\delta^2}}\Big] \\
&= \sum_{n=0}^{\infty} \frac{e^{-\lambda T}(\lambda T)^n}{n!} e^{-rT}K\mathbb{P}\Big[\xi < \frac{\ln(K/S_0) - (r - \lambda k + \frac{n\gamma}{T} - \frac{1}{2}(\sigma^2 + \frac{n\delta^2}{T}))T}{\sqrt{\sigma^2 + \frac{n\delta^2}{T}}\sqrt{T}}\Big] \\
&= \sum_{n=0}^{\infty} \frac{e^{-\lambda T}(\lambda T)^n}{n!} e^{-rT}K\mathbb{P}\Big[\xi < \frac{\ln(K/S_0) - (r - \lambda k + \frac{n\gamma}{T} - \frac{1}{2}\nu_n^2)T}{\nu_n\sqrt{T}}\Big] \\
&= \sum_{n=0}^{\infty} \frac{e^{-\lambda' T}(\lambda' T)^n}{n!} e^{-(r-\lambda k+\frac{n\gamma}{T})T}K\Phi\Big(\frac{\ln(K/S_0) - (r - \lambda k + \frac{n\gamma}{T} - \frac{1}{2}\nu_n^2)T}{\nu_n\sqrt{T}}\Big) \\
&= \sum_{n=0}^{\infty} \frac{e^{-\lambda' T}(\lambda' T)^n}{n!} e^{-r_n T}K\Phi\Big(\frac{\ln(K/S_0) - (r_n - \frac{1}{2}\nu_n^2)T}{\nu_n\sqrt{T}}\Big),
\end{aligned}$$

where $\lambda^{'} \equiv \lambda(1+k)$, $\gamma \equiv \ln(1+k)$, $r_n \equiv r - \lambda k + \frac{n\gamma}{T}$, and $\nu_n^2 \equiv \sigma^2 + \frac{n\delta^2}{T}$.

On the other hand, in order to evaluate B, we introduce a new measure $\hat{\mathbb{P}}$ equivalent to $\mathbb{P}$ by means of the Radon-Nikodym derivative

$$\frac{d\hat{\mathbb{P}}}{d\mathbb{P}} = \exp(\sigma W_T - \frac{1}{2}\sigma^2 T).$$

Then $\hat{W}_T = W_T - \sigma T$ follows a standard Brownian motion under the new measure $\hat{\mathbb{P}}$ by Girsanov Theorem. Now we find that

$$\begin{aligned}
B &= \mathbb{E}[e^{-rT} S_T I_D] \\
&= \hat{\mathbb{E}}\left[\frac{d\mathbb{P}}{d\hat{\mathbb{P}}} e^{-rT} S_T I_D\right] \\
&= \sum_{n=0}^{\infty} \frac{e^{-\lambda T}(\lambda T)^n}{n!} \cdot \\
&\quad \cdot \hat{\mathbb{E}}\left[\frac{d\mathbb{P}}{d\hat{\mathbb{P}}} e^{-rT} S_0 \exp\left(\sigma W_T + (r - \lambda k - \frac{1}{2}\sigma^2)T + \sum_{j=1}^{n} \ln(1+U_j)\right) I_D\right] \\
&= \sum_{n=0}^{\infty} \frac{e^{-\lambda T}(\lambda T)^n}{n!} \hat{\mathbb{E}}\left[S_0 \exp\left(-\lambda k T + \sum_{j=1}^{n} \ln(1+U_j)\right) I_D\right] \\
&= \sum_{n=0}^{\infty} \frac{e^{-\lambda T}(\lambda T)^n}{n!} e^{-\lambda k T + n\gamma} S_0 \, \hat{\mathbb{E}}[I_D]
\end{aligned}$$

where $I_D = I_{\{S_T < K\}}$ is an indicator function. Combining and simplifying the exponential functions yield

$$
\begin{aligned}
B &= \sum_{n=0}^{\infty} \frac{e^{-\lambda' T}(\lambda' T)^n}{n!} \; S_0 \, \hat{\mathbb{P}}[S_T < K] \\
&= \sum_{n=0}^{\infty} \frac{e^{-\lambda' T}(\lambda' T)^n}{n!} \; S_0 \cdot \\
&\qquad \cdot \hat{\mathbb{P}}\bigg[S_0 \exp\bigg(\sigma W_T + (r - \lambda k - \frac{1}{2}\sigma^2)T + \sum_{j=1}^{n} \ln(1 + U_j) \bigg) < K \bigg] \\
&= \sum_{n=0}^{\infty} \frac{e^{-\lambda' T}(\lambda' T)^n}{n!} \; S_0 \cdot \\
&\qquad \cdot \hat{\mathbb{P}}\bigg[S_0 \exp\bigg(\sigma W_T^Q + (r - \lambda k + \frac{1}{2}\sigma^2)T + \sum_{j=1}^{n} \ln(1 + U_j) \bigg) < K \bigg] \\
&= \sum_{n=0}^{\infty} \frac{e^{-\lambda' T}(\lambda' T)^n}{n!} \; S_0 \cdot \\
&\qquad \cdot \hat{\mathbb{P}}\bigg[\sigma \hat{W}_T + \sum_{j=1}^{n} \ln(1 + U_j) < \ln(K/S_0) - (r - \lambda k + \frac{1}{2}\sigma^2)T \bigg] \\
&= \sum_{n=0}^{\infty} \frac{e^{-\lambda' T}(\lambda' T)^n}{n!} \; S_0 \, \hat{\mathbb{P}}\bigg[\xi < \frac{\ln(K/S_0) - (r - \lambda k + \frac{1}{2}\sigma^2)T - nm}{\sqrt{\sigma^2 T + n\delta^2}} \bigg] \\
&= \sum_{n=0}^{\infty} \frac{e^{-\lambda' T}(\lambda' T)^n}{n!} \; S_0 \, \Phi\bigg(\frac{\ln(K/S_0) - (r_n + \frac{1}{2}\nu_n^2)T}{\nu_n \sqrt{T}} \bigg)
\end{aligned}
$$

This completes the alternative derivation of Merton's pricing formula for a European put option under jump-diffusion processes. $\square$

A.2 Proof of Theorem 4.1.3.

Proof. We prove this assertion by using mathematical induction. Now suppose that the formula holds for $n = m$:

$$\begin{aligned} V(S_m, m\Delta t) &= p(S_m, m\Delta t) \\ &+ (1-e^{-r\Delta t})K \sum_{k=1}^{m-1} \left[e^{-(m-k)r\Delta t} \int_0^{B_k} \psi(S_k, (m-k)\Delta t; S_m)\, dS_k \right] + o(\Delta t) \end{aligned}$$

Note that we have shown previously that for $n = 1, 2$, the above equation is satisfied. Now we obtain the following equation according to the value matching condition.

$$\begin{aligned} K - B_m &= p(B_m, m\Delta t) \\ &+ (1-e^{-r\Delta t})K \sum_{k=1}^{m-1} \left[e^{-(m-k)r\Delta t} \int_0^{B_k} \psi(S_k, (m-k)\Delta t; B_m)\, dS_k \right] + o(\Delta t) \end{aligned}$$

Also, according to Proposition 4.1.1, we obtain:

$$\begin{aligned} &V(S_{m+1}, (m+1)\Delta t) \\ &\quad = p(S_{m+1}, (m+1)\Delta t) \\ &\quad + \sum_{k=1}^{m} e^{-r(m+1-k)\Delta t} \int_0^{B_k} [K - S_k - V(S_k, \Delta t)]\psi(S_k, (m+1-k)\Delta t; S_{m+1}) dS_k \end{aligned}$$

The put-call parity and simple algebra give

$V(S_{m+1},(m+1)\Delta t)$

$$
\begin{aligned}
&= p(S_{m+1},(m+1)\Delta t) \\
&\quad + \Bigg\{ e^{-rm\Delta t} \int_0^{B_1} (1-e^{-r\Delta t}) K \psi(S_1, m\Delta t; S_{m+1}) dS_1 \\
&\qquad - e^{-rm\Delta t} \int_0^{B_1} c(S_1,\Delta t) \psi(S_1, m\Delta t; S_{m+1}) dS_1 \Bigg\} \\
&\quad + \Bigg\{ e^{-r(m-1)\Delta t} \int_0^{B_2} (1-e^{-2r\Delta t}) K \psi(S_2, (m-1)\Delta t; S_{m+1}) dS_2 \\
&\qquad - e^{-rm\Delta t} \int_0^{B_2} \int_0^{B_1} (1-e^{-r\Delta t}) K \psi(S_1, \Delta t; S_2) dS_1 \psi(S_2, (m-1)\Delta t; S_{m+1}) dS_2 \\
&\qquad - e^{-r(m-1)\Delta t} \int_0^{B_2} c(S_2, 2\Delta t) \psi(S_2, (m-1)\Delta t; S_{m+1}) dS_2 \Bigg\} \\
&\quad \vdots \\
&\quad + \Bigg\{ e^{-r\Delta t} \int_0^{B_m} (1-e^{-rm\Delta t}) K \psi(S_m, \Delta t; S_{m+1}) dS_m \\
&\qquad - e^{-r\Delta t} \int_0^{B_m} c(S_m, m\Delta t) \psi(S_m, \Delta t; S_{m+1}) dS_m \\
&\qquad - \sum_{k=1}^{m-1} e^{-r(m+1-k)\Delta t} \int_0^{B_m} \int_0^{B_k} (1-e^{-r\Delta t}) K \psi(S_k, (m-k)\Delta t; S_m) dS_k \\
&\qquad\qquad\qquad \psi(S_m, \Delta t; S_{m+1}) dS_m \Bigg\}
\end{aligned}
$$

By rearranging terms in the above equation, we obtain:

$V(S_{m+1},(m+1)\Delta t)$

$$
\begin{aligned}
&= p(S_{m+1},(m+1)\Delta t) \\
&\quad + \sum_{k=1}^{m} e^{-r(m+1-k)\Delta t} \int_0^{B_k} (1-e^{-r\Delta t}) K \psi(S_k, (m+1-k)\Delta t; S_{m+1}) dS_k \\
&\quad + L_1 + L_2 + \ldots + L_m,
\end{aligned}
$$

where

$$L_1 = -e^{-rm\Delta t} \int_0^{B_1} c(S_1, \Delta t)\psi(S_1, m\Delta t; S_{m+1})dS_1$$

$$L_2 = e^{-r(m)\Delta t} \int_0^{B_2} (1 - e^{-r\Delta t})K\psi(S_2, (m-1)\Delta t; S_{m+1})dS_2$$
$$- e^{-rm\Delta t} \int_0^{B_2} \int_0^{B_1} (1-e^{-r\Delta t})K\psi(S_1, \Delta t; S_2)dS_1\psi(S_2, (m-1)\Delta t; S_{m+1})dS_2$$
$$- e^{-r(m-1)\Delta t} \int_0^{B_2} c(S_2, 2\Delta t)\psi(S_2, (m-1)\Delta t; S_{m+1})dS_2$$
$$\vdots$$
$$L_m = e^{-2r\Delta t} \int_0^{B_m} (1 - e^{-r(m-1)\Delta t})K\psi(S_m, \Delta t; S_{m+1})dS_m$$
$$- e^{-r\Delta t} \int_0^{B_m} c(S_m, m\Delta t)\psi(S_m, \Delta t; S_{m+1})dS_m$$
$$- \sum_{k=1}^{m-1} e^{-r(m+1-k)\Delta t} \int_0^{B_m} \int_0^{B_k} (1 - e^{-r\Delta t})K\psi(S_k, (m-k)\Delta t; S_m)dS_k$$
$$\cdot \psi(S_m, \Delta t; S_{m+1})dS_m$$

Now we prove that L_i for $i = 1, 2, \cdots, m$ is of order higher than Δt.

Let $\hat{L}_i = -e^{r\Delta t}L_i$, for $i = 1, 2, \cdots, m$, then

$$\hat{L_m} = -e^{-r\Delta t} \int_0^{B_m} (1 - e^{-r(m-1)\Delta t})K\psi(S_m, \Delta t; S_{m+1})dS_m$$
$$+ \int_0^{B_m} c(S_m, m\Delta t)\psi(S_m, \Delta t; S_{m+1})dS_m$$
$$+ A_m,$$

where

$$A_m = (1 - e^{-r\Delta t})K \cdot$$
$$\sum_{k=1}^{m-1} \left[e^{-(m-k)r\Delta t} \int_0^{B_m} \int_0^{B_k} \psi(S_k, (m-k)\Delta t; S_m)\ dS_k \right] \psi(S_m, \Delta t; S_{m+1})dS_m$$

For $S_m < B_m$, it is satisfied that:

$$
\begin{aligned}
K - S_m \ &> \ V(S_m, m\Delta t) \\
&= p(S_m, m\Delta t) \\
&+ (1 - e^{-r\Delta t})K \sum_{k=1}^{m-1} \left[e^{-(m-k)r\Delta t} \int_0^{B_k} \psi(S_k, (m-k)\Delta t; S_m) dS_k \right] + o(\Delta t)
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
A_m &< \int_0^{B_m} [K - S_m - p(S_m, m\Delta t)] \ \psi(S_m, \Delta t; S_{m+1}) dS_m \\
&= (1 - e^{-mr\Delta t})K \int_0^{B_m} \ \psi(S_m, \Delta t; S_{m+1}) dS_{m+1} \\
&\quad - \int_0^{B_m} c(S_m, m\Delta t) \ \psi(S_m, \Delta t; S_{m+1}) dS_m
\end{aligned}
$$

By rearranging the terms, we obtain:

$$
\hat{L_m} < (1 - e^{-r\Delta t})K \int_0^{B_m} \psi(S_m, \Delta t; S_{m+1}) dS_m
$$

It is sufficient to prove that $(1 - e^{-r\Delta t})$ is of order Δt or higher, since the term, $\int_0^{B_m} \psi(S_m, \Delta t; S_{m+1}) dS_m$, is of order Δt or higher, which is proved in Proposition 2.

$$
1 - e^{-r\Delta t} = 1 - \sum_{n=0}^{\infty} \frac{(-r\Delta t)^n}{n!} = -\sum_{n=1}^{\infty} \frac{(-r\Delta t)^n}{n!}
$$

Thus L_i is of order higher than Δt. Therefore, we obtain the formula.

$$
\begin{aligned}
V(S_n, n\Delta t) &= p(S_n, n\Delta t) \\
&+ rK \sum_{k=1}^{n-1} \left[\Delta t \ e^{-(n-k)r\Delta t} \int_0^{B_k} \psi(S_k, (n-k)\Delta t; S_n) \ dS_k \right] + o(\Delta t)
\end{aligned}
$$

Since $o(\Delta t)$ converges to zero faster than the other two terms as Δt tends to zero, we obtain the value of American put price in the presence of jump in a continuous time with setting $n\Delta t = \tau$ and defining $S = S_n$:

$$V(S,\tau) = p(S,\tau) + rK \int_0^{\tau} e^{-r(\tau-s)} \left[\int_0^{B(s)} \psi(S_s, \tau - s; S)\, dS_s \right] ds,$$

which proves the assertion. $\square$

Appendix B

Tables and Figures

Table B.1: Comparison of American put prices [Case I]

This table provides American put prices computed in three different ways for several number of time steps (N), number of jump states (L), and the number of stock price meshed (M). The CPU time is an average value of running times computing the associated column. From now on, "New method" indicates a numerical procedure for American option prices using the extended integral equation method.
Parameters: $X = 100, S = (80, 90, 100, 110, 120), r = 6\%$, $\sigma = 0.3$, $\delta = 0.15$, $\lambda = 1.0$, $k = 0$. $\mathbb{E}U_1^2 = e^{\delta^2} - 1 = e^{0.05} - 1 = 0.02276$, $\sigma^2_{total} = \sigma^2 + \lambda \mathbb{E}U_1^2 = 0.11276$

	Jump-Diffusion case							Diffusion case
	American option						Euro	Amer
	(1)Ext. Integral Eqn Method		(2)Amin		(3)Zhang		Merton	Binomial
S	N=50	N=100	N=100 L=100	N=200 L=200	N=100 M=100	N=200 M=200		N=10000
	Maturity = 0.25 year							
80	20.062	20.059	20.055	20.055	20.048	20.052	19.292	20.082
90	11.678	11.676	11.678	11.674	11.662	11.669	11.380	11.782
100	5.922	5.920	5.915	5.918	5.896	5.910	5.812	6.048
110	2.656	2.654	2.656	2.655	2.646	2.651	2.617	2.722
120	1.097	1.096	1.092	1.094	1.094	1.095	1.083	1.089
CPU	38.0	149.5	39.8	631.8	25.7	207.5	< 1	127.6
std	0.1	1.4	0.2	1.9	0.2	0.2		
	Maturity = 1.0 year							
80	21.797	21.784	21.756	21.763	21.744	21.759	19.978	21.858
90	15.478	15.467	15.435	15.442	15.429	15.443	14.407	15.557
100	10.801	10.792	10.761	10.771	10.755	10.770	10.164	10.877
110	7.436	7.429	7.396	7.407	7.400	7.412	7.054	7.496
120	5.070	5.064	5.035	5.035	5.041	5.051	4.838	5.106
CPU	43.8	169.8	40.2	645.1	26.0	208.3	< 1	128.2
std	0.2	0.9	0.4	2.1	0.5	1.5		

(Continue)
MAC1 is the MacMillan-Zhang's analytical method, and MAC2 is the modified MacMillan-Zhang's method explained in Section 5.2, RIC3 is the three-point Richardson extrapolation method, and JU is the approximation method using three-piece exponential function.

	Numerical Method	Analytical Methods			
	Ext. Integral Eqn (N=200)	MAC1	MAC2	RIC3	JU
S_0	Maturity = 0.25 year				
80	20.057	20.039	20.144	20.075	20.065
90	11.674	11.669	11.727	11.637	11.667
100	5.919	5.936	5.947	5.925	5.911
110	2.654	2.674	2.669	2.658	2.650
120	1.096	1.111	1.103	1.095	1.094
	Maturity = 1.00 year				
80	21.778	21.710	21.943	21.745	21.834
90	15.462	15.450	15.580	15.348	15.495
100	10.788	10.827	10.875	10.725	10.807
110	7.426	7.493	7.490	7.417	7.437
120	5.062	5.140	5.109	5.075	5.068
CPU(sec)		< 1	< 1	< 1	2.2
RMSE		7.8E-3	7.0E-3	3.4E-3	1.6E-3

Table B.2: Comparison of American put prices [Case II]

This table provides American put prices computed in three different ways for several number of time steps (N), number of jump states (L), and the number of stock price meshed (M). The CPU time is an average value of running times computing the associated column.

Parameters: $S = 40, X = (30, 35, 40, 45, 50), r = 8\%$, $\sigma^2 = 0.05$, $\delta^2 = 0.05$, $\lambda = 5.0$, $k = 0$. $\mathbb{E}U_1^2 = e^{\delta^2} - 1 = e^{0.05} - 1 = 0.051271$, $\sigma^2_{total} = \sigma^2 + \lambda \mathbb{E}U_1^2 = 0.05 + (5.0)(0.051271) = 0.30635548$

	Jump-Diffusion case							Diffusion case
	American option						Euro	Amer
	(1)Ext. Integral Eqn Method		(2)Amin		(3)Zhang		Merton	Binomial
X	N=50	N=100	N=100 L=100	N=200 L=200	N=100 M=100	N=200 M=200		N=10000
	Maturity = 0.25 year							
30	0.675	0.675	0.673	0.674	0.675	0.674	0.670	0.639
35	1.689	1.688	1.688	1.688	1.690	1.688	1.673	1.876
40	3.631	3.630	3.632	3.630	3.628	3.629	3.592	4.043
45	6.736	6.734	6.736	6.734	6.734	6.734	6.655	7.117
50	10.700	10.698	10.698	10.697	10.698	10.697	10.545	10.933
CPU	26.3	105.4	28.6	452.3	18.6	147.5	< 1	120.7
std	0.8	0.7	0.8	0.7	0.8	1.3		
	Maturity = 1.0 year							
30	2.722	2.720	2.722	2.720	2.720	2.719	2.621	2.848
35	4.607	4.604	4.611	4.606	4.605	4.603	4.412	4.817
40	7.034	7.030	7.042	7.034	7.030	7.027	6.696	7.309
45	9.961	9.955	9.971	9.961	9.954	9.951	9.422	10.274
50	13.327	13.320	13.340	13.328	13.316	13.313	12.524	13.652
CPU	21.5	86.1	28.1	452.4	17.9	144.2	< 1	121.2
std	0.4	0.6	0.5	0.3	0.2	0.8		

(Continue)
MAC1 is the MacMillan-Zhang's analytical method, and MAC2 is the modified MacMillan-Zhang's method explained in Section 5.2, RIC3 is the three-point Richardson extrapolation method, and JU is the approximation method using three-piece exponential function.

	Numerical Method	Analytical Methods			
	Ext. Integral Eqn (N=100)	MAC1	MAC2	RIC3	JU
X	Maturity = 0.25 year				
30	0.6748	0.6844	0.6765	0.6743	0.6747
35	1.6881	1.7063	1.6922	1.6859	1.6880
40	3.6299	3.6608	3.6391	3.6235	3.6303
45	6.7339	6.7841	6.7494	6.7217	6.7353
50	10.6979	10.7721	10.7178	10.6741	10.7008
	Maturity = 1.00 year				
30	2.7200	2.7786	2.7446	2.7228	2.7109
35	4.6038	4.6760	4.6418	4.5996	4.5895
40	7.0295	7.1103	7.0834	7.0082	7.0098
45	9.9549	10.0397	10.0260	9.9059	9.9299
50	13.3202	13.4015	13.3958	13.2335	13.2917
CPU(sec)		< 1	< 1	< 1	2.2
RMSE		1.2E-2	5.7E-3	3.0E-3	2.0E-3

Table B.3: Comparison of American put prices [Case III]

This table provides American put prices computed in three different ways. The CPU time is an average value of running times computing the associated column.

Parameters:

$S = (40, 45, 50, 55)$, $X = 45$, $r = 9\%$, maturity = 0.25, 0.5, 1.0 year, $\sigma^2 = 0.004$, $k = 0$, $\lambda = 0.9$, $\mathbb{E}U_1^2 = 0.04$ ($\delta^2 = 0.039220713$), $\sigma^2_{total} = \sigma^2 + \lambda\mathbb{E}U_1^2 = 0.004 + (0.9)(0.04) = 0.04$

	Jump-Diffusion case							Diffusion case
	American option						Euro	Amer
	(1)Ext. Integral Eqn Method		(2)Amin		(3)Zhang		Merton	Binomial
Stock price	N=50	N=100	N=100 L=100	N=200 L=200	N=100 M=100	N=200 M=200		N=10000
	Maturity = 0.25 year							
40	5.000	5.000	5.000	5.000	5.000	5.000	4.376	5.000
45	0.855	0.855	0.671	0.803	0.823	0.848	0.810	1.416
50	0.349	0.349	0.193	0.301	0.349	0.349	0.330	0.233
55	0.166	0.166	0.054	0.127	0.166	0.166	0.157	0.023
CPU	41.6	163.4	35.5	563.7	22.6	182.2	< 1	120.8
std	0.3	1.3	0.3	2.7	0.1	2.5		
	Maturity = 1.0 year							
40	5.000	5.000	5.000	5.000	5.000	5.000	3.392	5.153
45	1.920	1.920	1.917	1.927	1.913	1.921	1.563	2.267
50	1.014	1.014	1.008	1.019	1.014	1.016	0.826	0.923
55	0.540	0.540	0.531	0.542	0.539	0.540	0.443	0.350
CPU	40.8	153.8	35.4	561.8	22.7	184.6	< 1	120.6
std	0.7	1.5	0.3	3.2	0.1	2.1		

(Continue)
MAC1 is the MacMillan-Zhang's analytical method, and MAC2 is the modified MacMillan-Zhang's method explained in Section 5.2, RIC3 is the three-point Richardson extrapolation method, and JU is the approximation method using three-piece exponential function.

	Numerical Method	Analytical Methods			
	Ext. Integral Eqn (N=200)	MAC1	MAC2	RIC3	JU
S_0	Maturity = 0.25 year				
40	5.0000	5.0000	5.0000	5.0000	5.0000
45	0.8556	1.0421	0.8688	0.8353	0.8610
50	0.3490	0.4174	0.3548	0.3412	0.3534
55	0.1658	0.1925	0.1690	0.1611	0.1681
	Maturity = 1.00 year				
40	5.0000	5.0000	5.0000	5.0000	5.0000
45	1.9210	2.3207	1.9598	1.8039	1.9775
50	1.0149	1.1965	1.0365	0.9407	1.0478
55	0.5402	0.6367	0.5514	0.4980	0.5570
CPU(sec)		< 1	< 1	< 1	2.2
RMSE		1.9E-1	1.9E-2	5.3E-2	2.3E-2

Table B.4: Comparison of American put prices varying k with constant volatility

This table provides American put prices computed in three different ways with varying k.

Parameters:

$S = 100$, $X = 100$, $r = 6\%$, maturity $= 0.25$, $\sigma = 0.3$, $\lambda = 1$, $\delta = 0.15$.

	(1)Ext. Integ. Eqn		(2)Amin		(3)Zhang			
k	N=100	N=200	N=100 L=100	N=200 L=200	N=100 M=100	N=200 M=200	$\mathbb{E}U_1^2$	σ^2_{total}
-0.9	19.798	19.798	1.619	3.103	1.855	1.870	0.810	0.900
-0.5	11.398	11.398	11.419	11.404	11.384	11.394	0.256	0.346
-0.1	6.190	6.189	6.183	6.193	6.165	6.179	0.028	0.118
0	5.920	5.919	5.915	5.918	5.896	5.910	0.023	0.113
0.0001	5.919	5.919	5.914	5.918	5.896	5.910	0.023	0.113
0.1	6.096	6.095	6.097	6.086	6.074	6.087	0.038	0.128
0.5	9.641	9.639	9.623	9.635	9.617	9.630	0.301	0.391
0.9	14.955	14.953	14.944	14.947	14.925	14.940	0.892	0.982

Table B.5: Comparison of American put prices varying k with constant total volatility

This table provides American put prices computed in three different ways with varying k.

Parameters:

$S = 100$, $X = 100$, $r = 6\%$, maturity $= 0.25$, $\lambda = 1$, $\delta = 0.15$, $\sigma^2_{total} = 0.9$.

	(1)Ext. Integ. Eqn		(2)Amin		(3)Zhang			
k	N=100	N=200	N=100 L=100	N=200 L=200	N=100 M=100	N=200 M=200	$\mathbb{E}U_1^2$	σ^2
-0.9	19.798	19.798	1.619	3.050	1.849	1.866	0.810	0.090
-0.89	19.722	19.721	1.938	15.042	3.016	2.976	0.792	0.108
-0.88	19.668	19.667	2.357	19.559	5.229	5.101	0.775	0.125
-0.87	19.629	19.627	4.597	19.635	8.446	8.246	0.757	0.143
-0.86	19.600	19.598	11.875	19.613	11.963	11.743	0.740	0.160
-0.85	19.578	19.576	17.950	19.586	14.971	14.785	0.723	0.177
-0.8	19.511	19.509	19.544	19.532	19.419	19.413	0.641	0.259
-0.7	19.327	19.325	19.348	19.332	19.311	19.318	0.492	0.408
-0.6	18.994	18.634	19.029	18.865	18.972	18.981	0.364	0.536
-0.5	18.636	18.634	18.668	18.648	18.610	18.621	0.256	0.644
-0.1	18.030	18.027	18.022	18.018	17.990	18.010	0.028	0.872
0	17.992	17.989	17.947	17.951	17.951	17.968	0.023	0.877
0.0001	17.992	17.989	17.947	17.949	17.951	17.968	0.023	0.877
0.1	17.938	17.935	17.895	17.904	17.897	17.915	0.038	0.862
0.5	16.959	16.956	16.959	16.960	16.927	16.941	0.301	0.599
0.9	14.547	14.541	14.531	14.539	14.516	14.525	0.892	0.008

Table B.6: American put prices with respect to jump intensity

This table provides American put prices with varying the ratio of the variance of diffusion component to the total variance ($\sigma_D^2/\sigma_{total}^2$). The binomial price is computed with 10,000 number of time steps.

Parameters:

$S = 40$, $X = (30, 50)$, $r = 8\%$, maturity $= 0.25$ year,

$\lambda = (5, 3, 1, 0.5, 0.2, 0.1, 0.01, 0.0001)$, $\delta^2 = 0.05$, $k = 0$, $\sigma_{total}^2 = 0.306355$.

	(1)Ext. Integ. Eqn(N=100)								Binomial
$\frac{\sigma_D^2}{\sigma_{total}^2}$	0.163	0.498	0.833	0.916	0.967	0.983	0.998	0.999	1.000
λ	5	3	1	0.5	0.2	0.1	0.01	0.0001	0
σ^2	0.0500	0.1525	0.2551	0.2807	0.2961	0.3012	0.3058	0.30635	0.306355
X=30	0.6748	0.6415	0.6360	0.6373	0.6385	0.6389	0.6393	0.6393	0.6390
X=50	10.6979	10.7736	10.8816	10.9086	10.9246	10.9298	10.9344	10.9350	10.9329
	(2)Amin's Method (N=L=200)								Binomial
X=30	0.6740	0.6398	0.6356	0.6373	0.6359	0.6367	0.6383	0.6385	0.6390
X=50	10.6972	10.7691	10.8809	10.9084	10.9238	10.9286	10.9330	10.9334	10.9329
	(3)Zhang's Method (N=M=100)								Binomial
X=30	0.6744	0.6414	0.6356	0.6367	0.6377	0.6380	0.6384	0.6384	0.6390
X=50	10.6968	10.7711	10.8767	10.9027	10.9190	10.9230	10.9276	10.9281	10.9329

Table B.7: Comparison of Running time

This table provides American put prices computed in three different ways with varying the number of time steps (N), the number of jump states (L), and the number of stock price meshed (M). The CPU time is an average value of running times computing the associated column.

Parameters:

$S = 40$, $X = (30, 50)$, $r = 8\%$, $\sigma^2 = 0.05$, maturity = 0.25 year,
$\delta^2 = 0.05$, $\lambda = 5.0$, $k = 0$.

	(1) Ext. Integ. Eqn Method						
	N=5	N=10	N=20	N=40	N=80	N=160	N=320
X=30	0.6814	0.6779	0.6761	0.6752	0.6748	0.6746	0.6746
X=50	10.7522	10.7235	10.7079	10.7012	10.6983	10.6970	10.6964
CPU(sec)	0.8	2.4	8.0	30.2	116.2	463.1	1873.7
	(2-1) Amin's Method						
	N=5	N=10	N=20	N=40	N=80	N=160	N=320
	L=5	L=10	L=20	L=40	L=80	L=160	L=320
X=30	0.0885	0.2801	0.5286	0.6575	0.6730	0.6738	0.6742
X=50		10.0000	10.3852	10.6731	10.6987	10.6974	10.6968
CPU(sec)		0.0	0.1	1.4	20.4	314.7	5022.1
	(2-2) Amin's Method (Truncation method)						
	N=5	N=10	N=20	N=40	N=80	N=160	N=320
	L=14	L=20	L=27	L=38	L=54	L=76	L=108
X=30	0.6312	0.6507	0.6476	0.6493	0.6511	0.6504	0.6514
X=50	10.7042	10.6864	10.6646	10.6555	10.6544	10.6499	10.6506
CPU(sec)	0.0	0.0	0.1	1.4	7.3	57.8	442.4
	(3) Zhang's Method						
	N=20	N=40	N=80	N=80	N=160	N=80	N=320
	M=20	M=40	M=80	M=160	M=160	M=320	M=320
X=30	0.6786	0.6768	0.6752	0.6740	0.6742	0.6740	0.6744
X=50	10.7201	10.7072	10.6975	10.6974	10.6967	10.6974	10.6966
CPU(Sec)	0.1	1.1	8.4	35.0	67.1	134.9	541.0

Table B.8: American call prices with dividends

This table provides American call prices with varying dividend rates.

Parameters:

$r = 8\%$, $\sigma^2 = 0.05$, $\delta^2 = 0.05$, and $k = 0$.

dividend	European	American option		
		N=20	N=50	N=100
	Maturity=1.00, S=50, X=50, λ=1.0			
0.001	7.9318	7.9326	7.9321	7.9318
0.01	7.4169	7.4250	7.4201	7.4185
0.1	3.4460	3.9369	3.9119	3.9036
0.2	1.2292	2.7270	2.6967	2.6867
0.3	0.3848	2.3166	2.2790	2.2664
	Maturity=1.00, S=50, X=50, λ=5.0			
0.01	11.6797	11.6878	11.6829	11.6813
0.1	7.5609	8.1140	8.1140	8.1052
0.2	4.3587	6.0904	6.0578	6.0467
0.3	2.3464	5.0061	4.9690	4.9561
	Maturity=1.00, S=50, X=40, λ=1.0			
0.01	16.5445	16.5543	16.5484	16.5464
0.1	11.4949	12.5442	12.5139	12.5038
0.2	7.1749	10.5066	10.4685	10.4524
0.3	4.1700	9.7565	9.81712	9.8416
	Maturity=1.00, S=40, X=50, λ=1.0			
0.01	6.0410	6.0458	6.0429	6.0419
0.1	3.6414	3.8815	3.8655	3.8602
0.2	1.9453	2.6443	2.6256	2.6193
0.3	0.9751	2.0120	1.9925	1.9859

Table B.9: Option prices with Bivariate distribution for jump size

This table provides the prices of put options when the jump size follows a bivariate distribution. The option prices corresonding to Amin's method are from Table II of Amin (1993).

Parameters:

$S = 40, X = (30, 35, 40, 45, 50), r = 8\%$, annual variance of diffusion component(σ^2) $= 0.05$, $\delta^2 = 0.05$, $\lambda = 5.0$. Probability that the log of the proportional jump magnitude is $= \delta$ is 0.5 and that it is $-\delta$ is 0.5.

$\mathbb{E}U_1^2 = 0.5(e^{\delta} - 1)^2 + 0.5(e^{-\delta} - 1)^2 = 0.05147$,

$\sigma^2_{total} = \sigma^2 + \lambda \mathbb{E}U_1^2 = 0.05 + (5.0)(0.0515) = 0.3075$.

Note that $\delta = \xi$, and $k = \mathbb{E}U_1 = 0.5(e^{\xi} - 1) + 0.5(e^{-\xi} - 1) = 0.0251$ in this case.

	Amin		Ext. Integ. Eqn	
	European	American	European	American
Exercise Price		N=100		N=20
	Maturity=0.25 year			
X=30	0.596	0.599	0.608	0.614
35	1.639	1.660	1.745	1.764
40	3.593	3.650	3.820	3.872
45	6.495	6.639	6.874	6.988
50	10.318	10.576	10.635	10.857
	Maturity=1.0 year			
X=30	2.410	2.510	2.652	2.761
35	4.142	4.354	4.494	4.705
40	6.370	6.752	6.819	7.185
45	9.045	9.665	9.569	10.148
50	12.105	13.035	12.676	13.536
CPU (sec)				91.5

Table B.10: Option prices with double-exponential distribution for jump size
Parameters:
$S = 40, X = (30, 35, 40, 45, 50), r = 8\%$, $\sigma^2 = 0.05$, $T = (0.25, 1)$ year $\delta^2 = 0.05$, $\lambda = 5.0$, $\mathbb{E}U_1 = 0$.
The corresponding paremeters, p, q, η_1, and η_2, for the double-exponential distribution are $p = 0.8045, q = 0.1955, \eta_1 = 15.6405$, and $\eta_2 = 2.5576$.

T=0.25 year	Euro call	Euro Put	Euro Put(Merton)
X=30	11.4904	0.8964	0.6697
35	7.2438	1.5509	1.6727
40	3.7377	2.9456	3.5920
45	1.5965	5.7054	6.6547
50	0.6291	9.6390	10.5445
T=1 year	Euro call	Euro Put	Euro Put(Merton)
X=30	15.0939	2.7874	2.6211
35	11.9118	4.2209	4.4116
40	9.1505	6.0752	6.6959
45	6.8544	8.3946	9.4222
50	5.0266	11.1824	12.5238

Table B.11: Put prices with respect to η [Case I]

This table provides European and American put prices varying η. The American put prices are computed using Equation (4.7).

Parameters: $S = (80, 90, 100, 110, 120), K = (30, 35, 40, 45, 50), r = 6\%$, $\sigma^2 = 0.3$, $\delta^2 = 0.15$, $\lambda = 1.0$, $k = 0$. $\mathbb{E}U_1^2 = 0.02276$, $\sigma_{total}^2 = 0.11276$.

η	1.0	0.5	0.2	0	-0.2	-0.4	-0.6	-0.8	-1.0
	European put prices								
S	Maturity = 0.25 year								
80	19.247	19.269	19.283	19.292	19.301	19.311	19.320	19.329	19.338
90	11.337	11.358	11.371	11.380	11.389	11.398	11.407	11.416	11.425
100	5.801	5.807	5.810	5.812	5.815	5.817	5.820	5.823	5.825
110	2.637	2.627	2.621	2.617	2.613	2.609	2.605	2.601	2.597
120	1.115	1.099	1.089	1.083	1.076	1.069	1.063	1.056	1.049
	Maturity = 1 year								
80	19.917	19.947	19.966	19.978	19.991	20.004	20.017	20.030	20.042
90	14.361	14.383	14.397	14.407	14.417	14.426	14.436	14.446	14.456
100	10.140	10.152	10.159	10.164	10.170	10.175	10.180	10.186	10.191
110	7.051	7.052	7.053	7.054	7.054	7.055	7.056	7.057	7.058
120	4.852	4.845	4.841	4.838	4.835	4.832	4.830	4.827	4.824
	American put prices								
K	Maturity = 0.25 year								
80	20.051	20.057	20.061	20.062	20.065	20.067	20.069	20.070	20.074
90	11.654	11.666	11.674	11.678	11.683	11.689	11.694	11.700	11.705
100	5.920	5.921	5.921	5.922	5.922	5.923	5.923	5.924	5.925
110	2.681	2.668	2.661	2.656	2.651	2.646	2.641	2.636	2.631
120	1.131	1.114	1.104	1.097	1.089	1.082	1.075	1.068	1.061
	Maturity = 1 year								
80	21.773	21.784	21.792	21.797	21.802	21.807	21.812	21.817	21.823
90	15.461	15.469	15.474	15.478	15.482	15.485	15.489	15.493	15.497
100	10.799	10.800	10.801	10.801	10.802	10.802	10.803	10.804	10.805
110	7.450	7.443	7.439	7.436	7.433	7.430	7.428	7.425	7.423
120	5.096	5.083	5.075	5.070	5.064	5.059	5.053	5.048	5.043

Table B.12: Put prices with respect to η [Case II]

This table provides European and American put prices varying η. The American put prices are computed using Equation (4.7).

Parameters: $S = 40, K = (30, 35, 40, 45, 50), r = 8\%$, $\sigma^2 = 0.05$, $\delta^2 = 0.05$, $\lambda = 5.0$, $k = 0$. $\mathbb{E}U_1^2 = 0.051271$, $\sigma^2_{total} = 0.30635548$

η	1.0	0.5	0.2	0	-0.2	-0.4	-0.6	-0.8	-1.0
	European put prices								
K	Maturity = 0.25 year								
30	0.747	0.712	0.687	0.670	0.651	0.631	0.610	0.589	0.566
35	1.767	1.720	1.692	1.673	1.654	1.636	1.619	1.604	1.590
40	3.557	3.563	3.578	3.592	3.610	3.632	3.658	3.686	3.718
45	6.403	6.517	6.598	6.655	6.714	6.774	6.835	6.895	6.955
50	10.210	10.375	10.477	10.544	10.611	10.677	10.740	10.802	10.861
	Maturity = 1 year								
30	2.618	2.618	2.619	2.621	2.624	2.627	2.632	2.638	2.644
35	4.342	4.371	4.394	4.412	4.431	4.452	4.476	4.500	4.527
40	6.540	6.608	6.659	6.696	6.736	6.778	6.822	6.867	6.914
45	9.174	9.286	9.365	9.422	9.482	9.543	9.606	9.669	9.732
50	12.193	12.346	12.450	12.524	12.599	12.676	12.753	12.830	12.905
	American put prices								
K	Maturity = 0.25 year								
30	0.757	0.719	0.694	0.675	0.656	0.635	0.614	0.592	0.569
35	1.794	1.741	1.710	1.689	1.668	1.649	1.631	1.614	1.600
40	3.620	3.613	3.621	3.631	3.646	3.665	3.687	3.713	3.742
45	6.525	6.617	6.686	6.736	6.788	6.843	6.898	6.954	7.009
50	10.437	10.562	10.643	10.700	10.756	10.813	10.867	10.921	10.973
	Maturity = 1 year								
30	2.752	2.734	2.726	2.722	2.719	2.717	2.717	2.718	2.720
35	4.592	4.592	4.599	4.607	4.617	4.629	4.644	4.660	4.679
40	6.960	6.985	7.012	7.034	7.059	7.088	7.118	7.152	7.186
45	9.827	9.879	9.924	9.961	10.000	10.042	10.085	10.132	10.179
50	13.148	13.221	13.281	13.327	13.376	13.425	13.478	13.532	13.586

Table B.13: Put prices with respect to η [Case III]

This table provides European and American put prices varying η. The American put prices are computed using Equation (4.7).

Parameters: $S = (40, 45, 50, 55)$, $X = 45$, $r = 9\%$, $\sigma^2 = 0.004$, $k = 0$, $\lambda = 0.9$, $\mathbb{E}U_1^2 = 0.04$ ($\delta^2 = 0.039220713$), $\sigma_{total}^2 = 0.04$.

η	1.0	0.5	0.2	0	-0.2	-0.4	-0.6	-0.8	-1.0
				European put prices					
S				Maturity = 0.25 year					
45	0.865	0.835	0.819	0.810	0.802	0.794	0.788	0.782	0.778
50	0.401	0.366	0.345	0.330	0.315	0.300	0.285	0.269	0.253
55	0.194	0.176	0.165	0.156	0.148	0.140	0.131	0.122	0.113
				Maturity = 0.5 year					
45	1.286	1.231	1.201	1.182	1.165	1.149	1.134	1.121	1.110
50	0.656	0.609	0.579	0.558	0.536	0.514	0.491	0.467	0.444
55	0.333	0.307	0.289	0.276	0.263	0.250	0.236	0.222	0.208
				Maturity = 1 year					
45	1.688	1.626	1.588	1.563	1.539	1.515	1.493	1.472	1.453
50	0.931	0.882	0.850	0.826	0.802	0.776	0.750	0.722	0.693
55	0.515	0.482	0.459	0.443	0.426	0.408	0.390	0.371	0.351
				American put prices					
S				Maturity = 0.25 year					
45	0.921	0.886	0.866	0.855	0.844	0.834	0.825	0.817	0.810
50	0.426	0.388	0.364	0.349	0.332	0.315	0.299	0.282	0.265
55	0.208	0.188	0.174	0.166	0.156	0.147	0.138	0.128	0.118
				Maturity = 0.5 year					
45	1.446	1.376	1.338	1.312	1.288	1.268	1.246	1.227	1.210
50	0.742	0.683	0.646	0.620	0.594	0.568	0.541	0.513	0.486
55	0.378	0.344	0.323	0.307	0.292	0.276	0.260	0.243	0.227
				Maturity = 1 year					
45	2.127	2.023	1.961	1.920	1.881	1.842	1.806	1.770	1.737
50	1.175	1.099	1.049	1.014	0.978	0.942	0.905	0.866	0.827
55	0.645	0.595	0.563	0.540	0.516	0.492	0.467	0.441	0.415

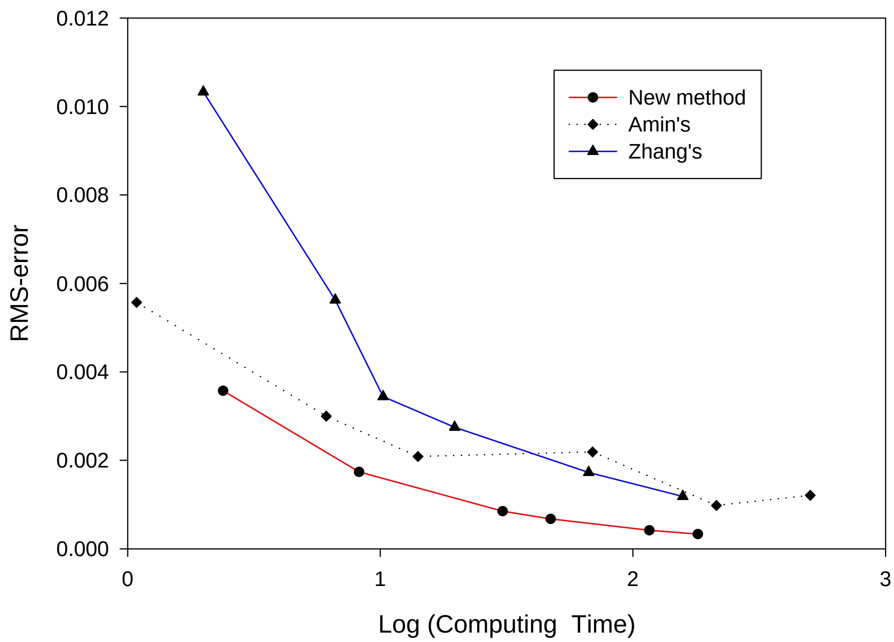


Figure B.1: Comparison of Efficiency for American-put prices: [Case A] short-time-to-maturity and small λ

In the figure, "New method" indicates a numerical procedure for American option prices using the extended integral equation method. *Parameters*:
Number of samples $(n) = 10$
DATA SET I: $S = 40$, $X = 30, 35, 40, 45, 50$, $r = 8\%$, maturity=0.25 year, $\lambda = 0.0001$, $\delta^2 = 0.05$, $k = 0$, $\sigma^2 = 0.30635035$, $\sigma^2_{total} = \sigma^2 + \lambda \mathbb{E} U_1^2 = 0.30635548$.

DATA SET II: $X = 100, S = (80, 90, 100, 110, 120), r = 6\%$, maturity=0.25 year, $\lambda = 0.0001$, $\delta = 0.15$, $k = 0$, $\sigma^2 = 0.11275$, $\sigma^2_{total} = \sigma^2 + \lambda \mathbb{E} U_1^2 = 0.11276$

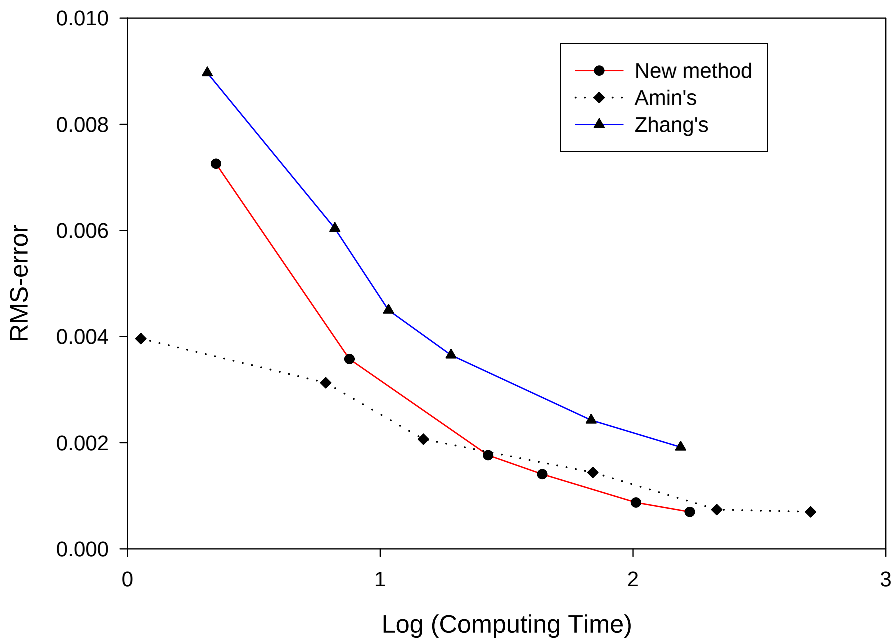


Figure B.2: Comparison of Efficiency for American-put prices: [Case B] long-time-to-maturity and small λ

Parameters:
Number of samples $(n) = 10$
DATA SET I: $S = 40$, $X = 30, 35, 40, 45, 50$, $r = 8\%$, maturity=1.0 year, $\lambda = 0.0001$, $\delta^2 = 0.05$, $k = 0$, $\sigma^2 = 0.30635035$, $\sigma^2_{total} = \sigma^2 + \lambda \mathbb{E} U_1^2 = 0.30635548$.

DATA SET II: $X = 100, S = (80, 90, 100, 110, 120), r = 6\%$, maturity=1.0 year, $\lambda = 0.0001$, $\delta = 0.15$, $k = 0$, $\sigma^2 = 0.11275$, $\sigma^2_{total} = \sigma^2 + \lambda \mathbb{E} U_1^2 = 0.11276$

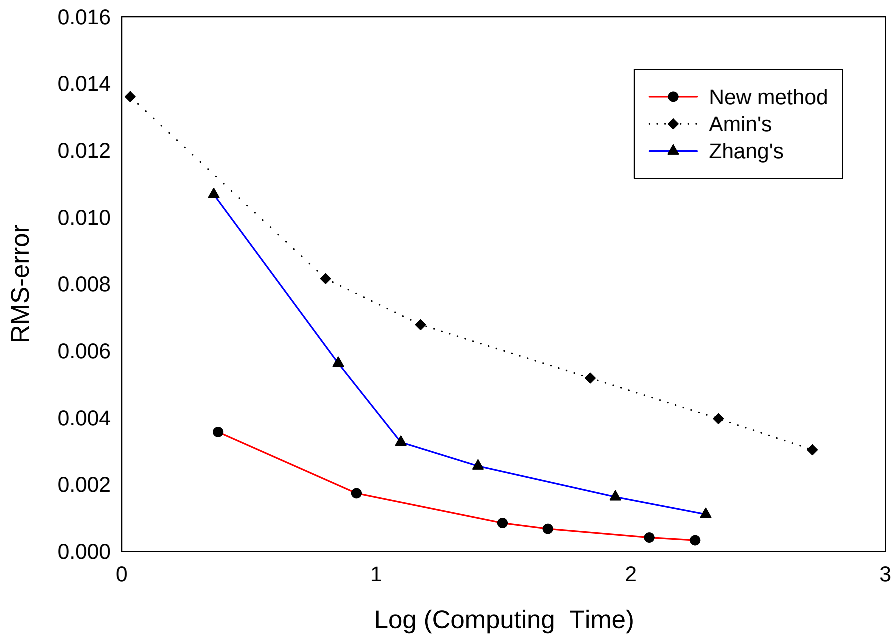


Figure B.3: Comparison of Efficiency for American-put prices: [Case C] short-time-to-maturity and small volatility of jump size

Parameters:
Number of samples $(n) = 10$
DATA SET I: $S = 40$, $X = 30, 35, 40, 45, 50$, $r = 8\%$, maturity=0.25 year, $\lambda = 5.0$, $\delta^2 = 0.0001$, $k = 0$, $\sigma^2 = 0.30585548$, $\sigma^2_{total} = \sigma^2 + \lambda \mathbb{E} U_1^2 = 0.30635548$.

DATA SET II: $X = 100, S = (80, 90, 100, 110, 120), r = 6\%$, maturity=0.25 year, $\lambda = 1.0$, $\delta = 0.01$, $k = 0$, $\sigma^2 = 0.11266$, $\sigma^2_{total} = \sigma^2 + \lambda \mathbb{E} U_1^2 = 0.11276$

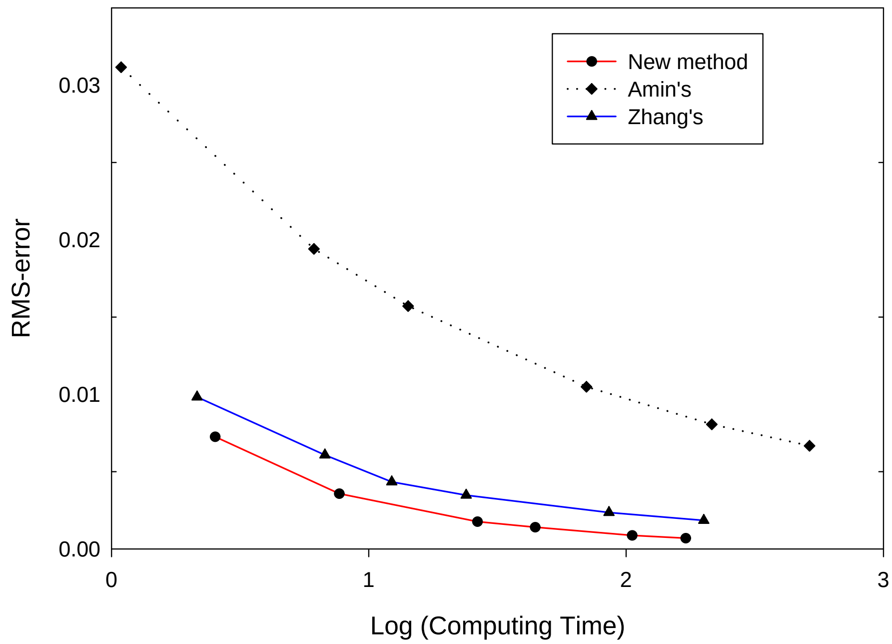


Figure B.4: Comparison of Efficiency for American-put prices: [Case D] long-time-to-maturity small volatility of jump size

Parameters:
Number of samples $(n) = 10$
DATA SET I: $S = 40$, $X = 30, 35, 40, 45, 50$, $r = 8\%$, maturity=1.0 year, $\lambda = 5.0$, $\delta^2 = 0.0001$, $k = 0$, $\sigma^2 = 0.30585548$, $\sigma^2_{total} = \sigma^2 + \lambda \mathbb{E} U_1^2 = 0.30635548$.

DATA SET II: $X = 100, S = (80, 90, 100, 110, 120), r = 6\%$, maturity=1.0 year, $\lambda = 1.0$, $\delta = 0.01$, $k = 0$, $\sigma^2 = 0.11266$, $\sigma^2_{total} = \sigma^2 + \lambda \mathbb{E} U_1^2 = 0.11276$

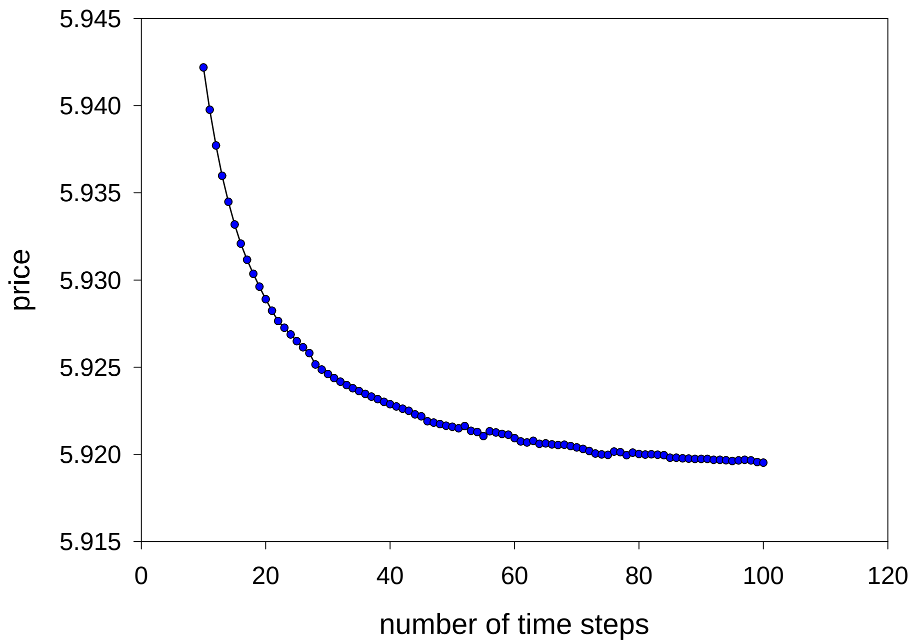


Figure B.5: Prices varying the number of time steps using Ext. integral eqn method [Case IA]

Parameters: S=100, X=100, r=6%, σ = 0.3, δ = 0.15, λ = 1.0, k = 0, maturity=0.25 year.

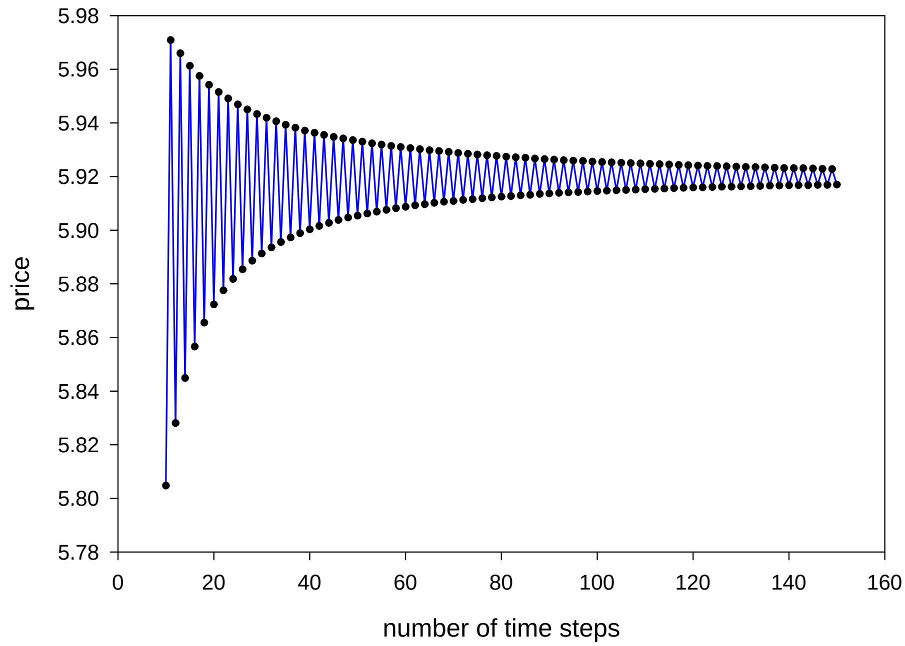


Figure B.6: Prices varying the number of time steps using Amin's method [Case IA]

Parameters: S=100, X=100, r=6%, $\sigma = 0.3$, $\delta = 0.15$, $\lambda = 1.0$, k = 0, maturity=0.25 year.

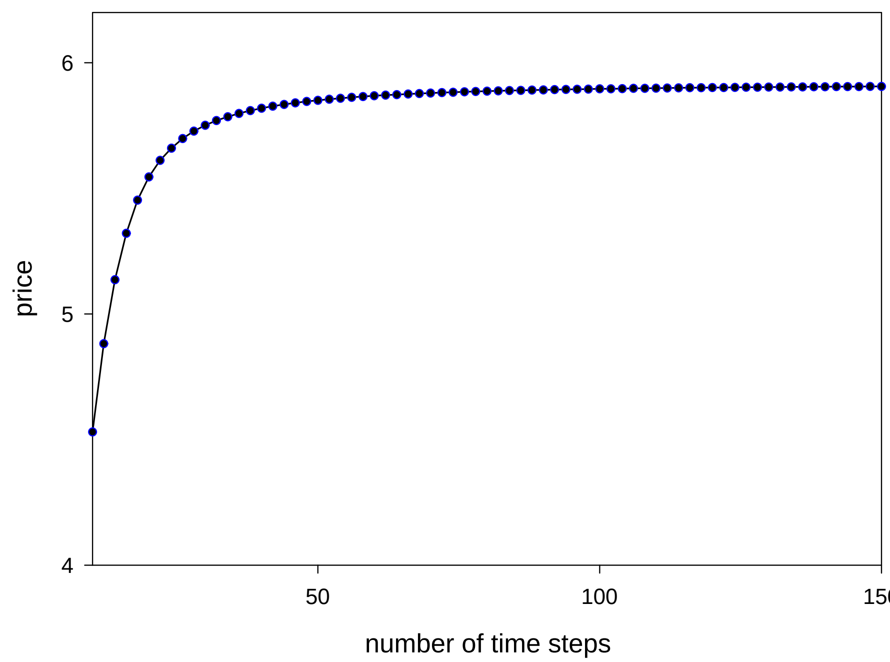


Figure B.7: Prices varying the number of time steps using Zhang's method [Case IA]

Parameters: S=100, X=100, r=6%, $\sigma = 0.3$, $\delta = 0.15$, $\lambda = 1.0$, k = 0, maturity=0.25 year.

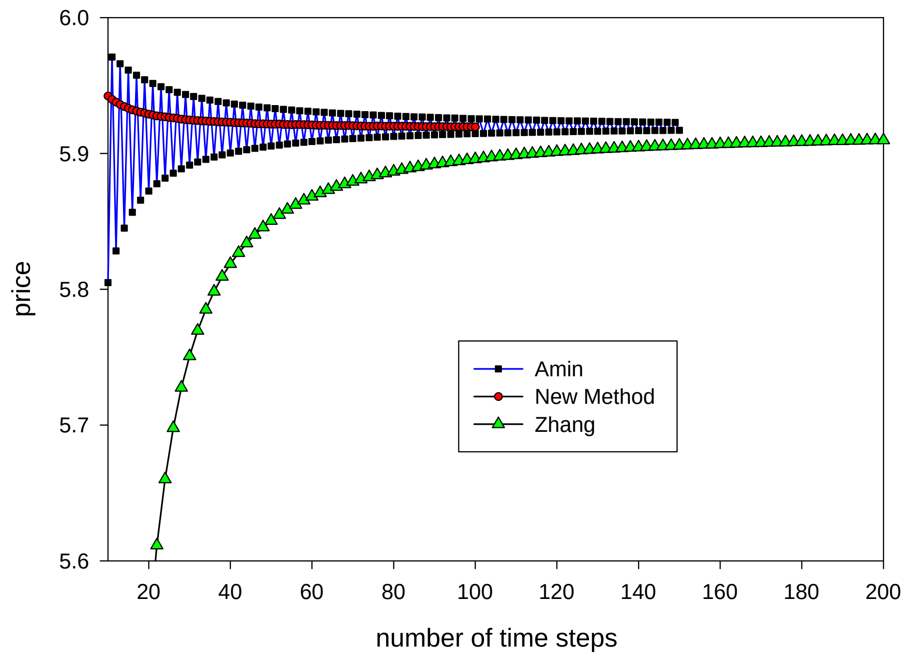


Figure B.8: Prices varying the number of time steps [Case IA]

Parameters: S=100, X=100, r=6%, σ = 0.3, δ = 0.15, λ = 1.0, k = 0, maturity=0.25 year.

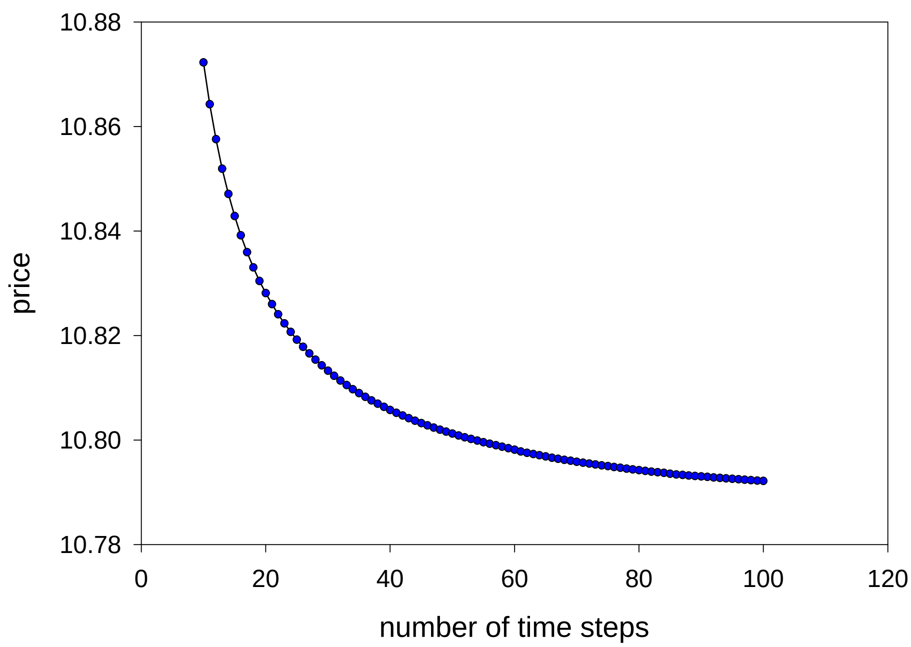


Figure B.9: Prices varying the number of time steps using Ext. integral eqn method [Case IB]

Parameters: S=100, X=100, r=6%, σ = 0.3, δ = 0.15, λ = 1.0, k = 0, maturity=1.0 year.

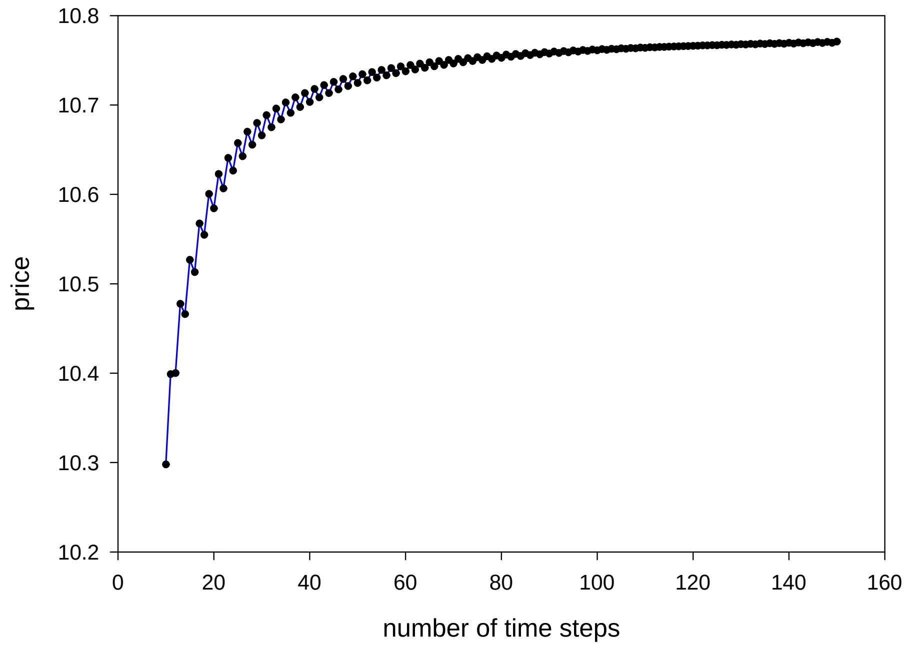


Figure B.10: Prices varying the number of time steps using Amin's method [Case IB]

Parameters: S=100, X=100, r=6%, σ = 0.3, δ = 0.15, λ = 1.0, k = 0, maturity=1.0 year.

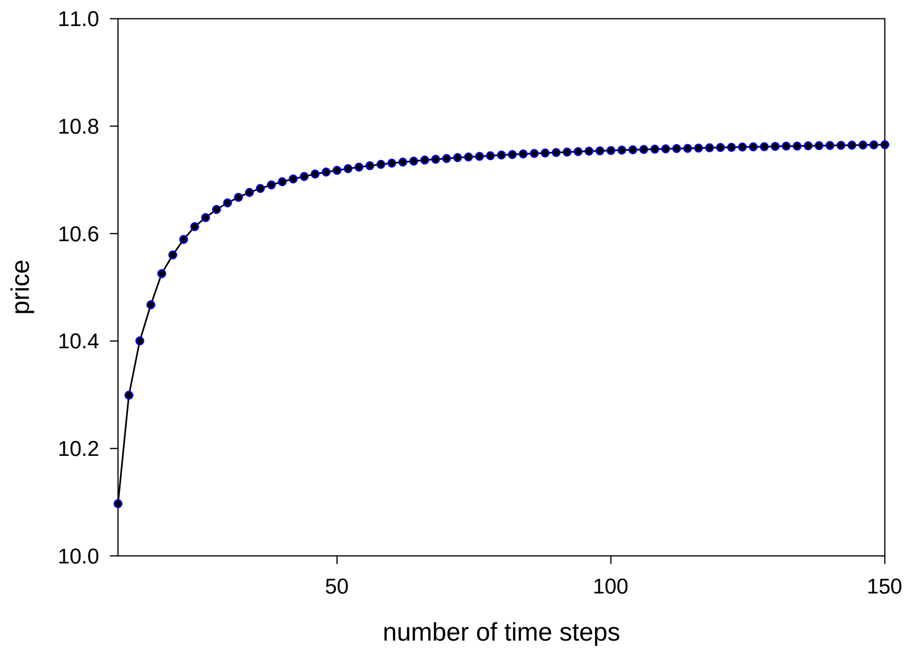


Figure B.11: Prices varying the number of time steps using Zhang's method [Case IB]

Parameters: S=100, X=100, r=6%, $\sigma = 0.3$, $\delta = 0.15$, $\lambda = 1.0$, k = 0, maturity=1.0 year.

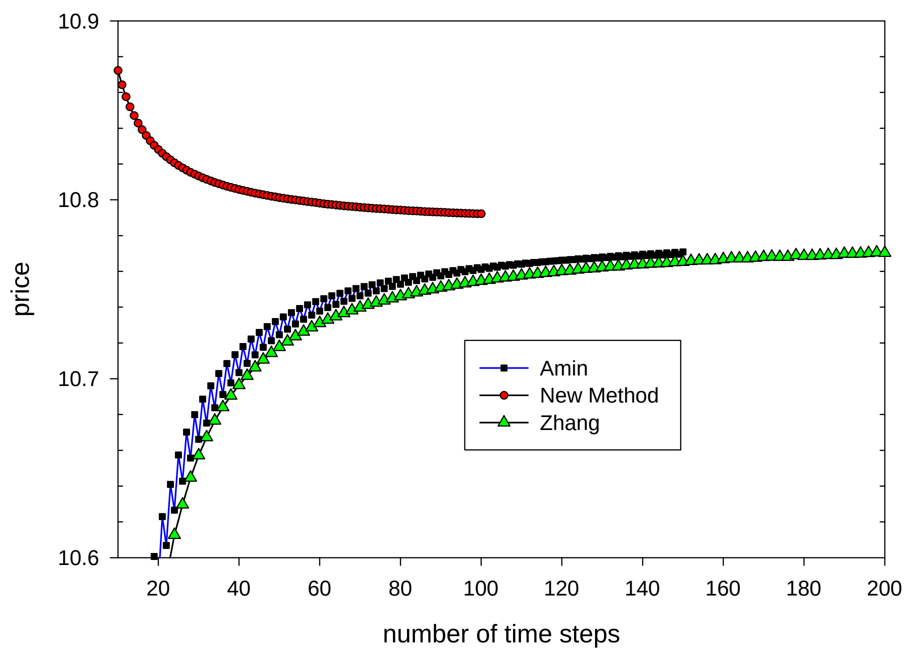


Figure B.12: Prices varying the number of time steps [Case IB]

Parameters: S=100, X=100, r=6%, $\sigma = 0.3$, $\delta = 0.15$, $\lambda = 1.0$, k $= 0$, maturity=1.0 year.

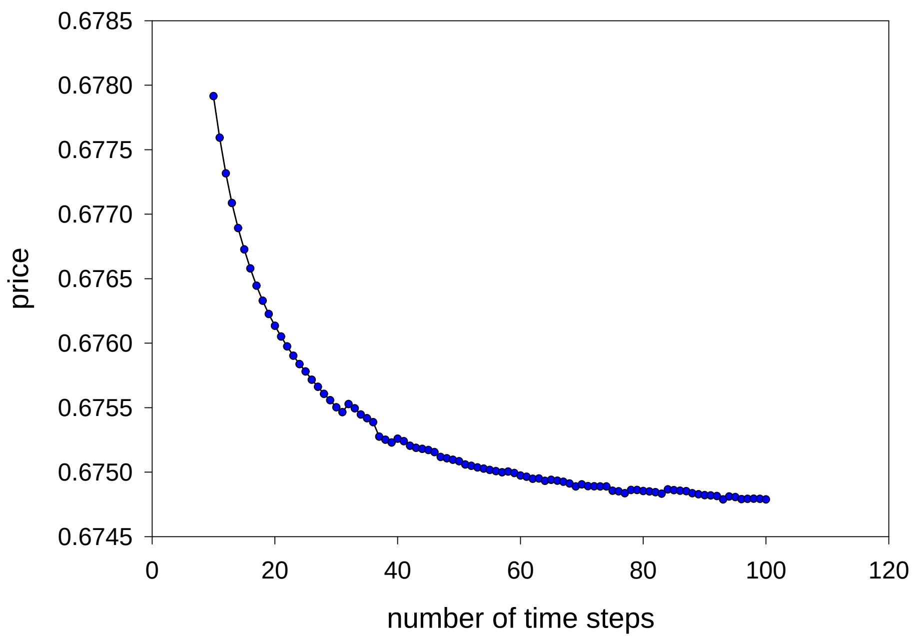


Figure B.13: Prices varying the number of time steps using Ext. integral eqn method [Case IIA]

Parameters: S=40, X=30, r=8%, $\sigma^2 = 0.05$, $\delta^2 = 0.05$, $\lambda = 5.0$, k = 0, maturity=0.25 year.

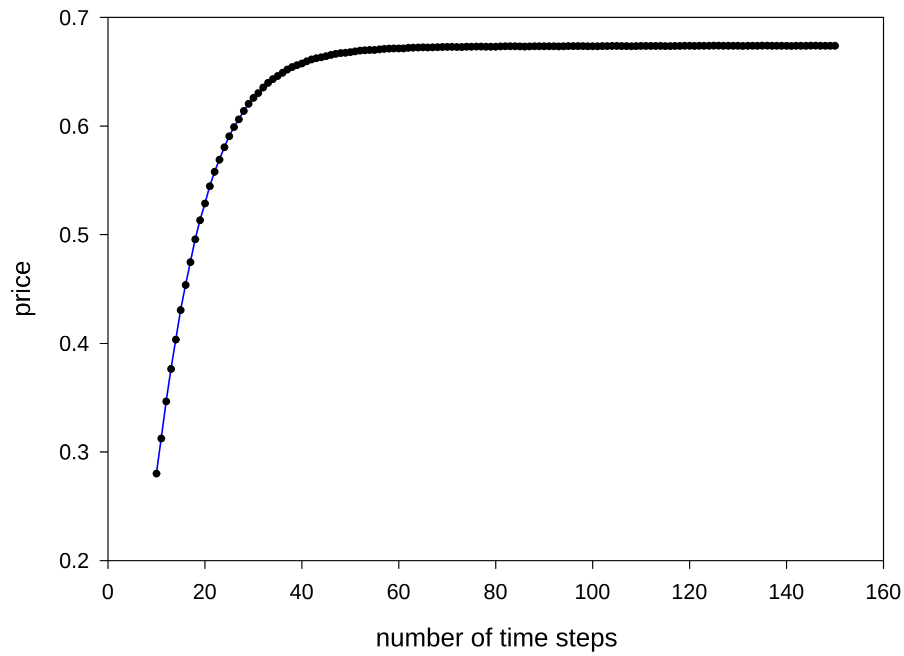


Figure B.14: Prices varying the number of time steps using Amin's method [Case IIA]

Parameters: S=40, X=30, r=8%, $\sigma^2 = 0.05$, $\delta^2 = 0.05$, $\lambda = 5.0$, k = 0, maturity=0.25 year.

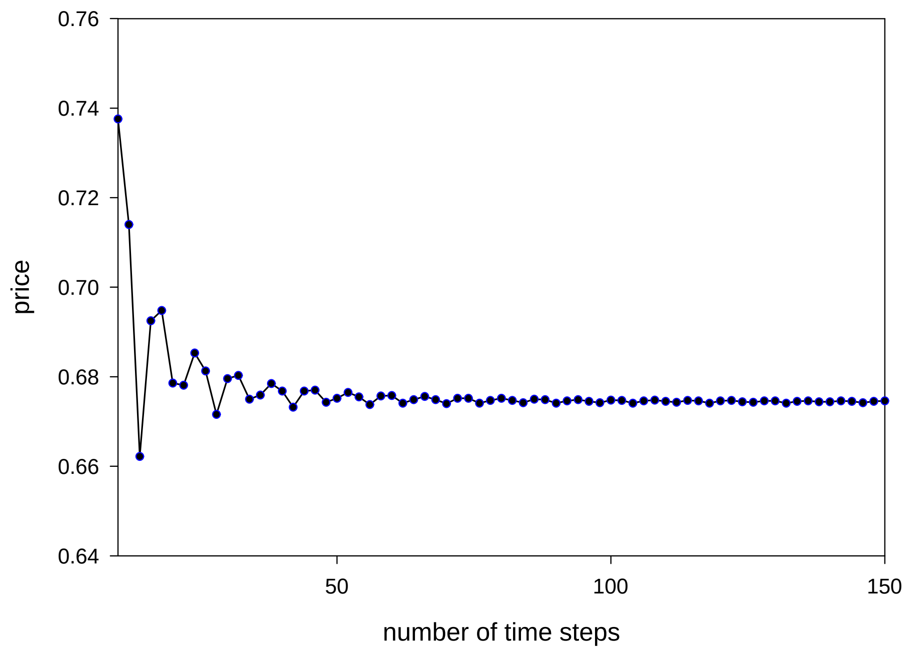


Figure B.15: Prices varying the number of time steps using Zhang's method [Case IIA]

Parameters: S=40, X=30, r=8%, $\sigma^2 = 0.05$, $\delta^2 = 0.05$, $\lambda = 5.0$, k = 0, maturity=0.25 year.

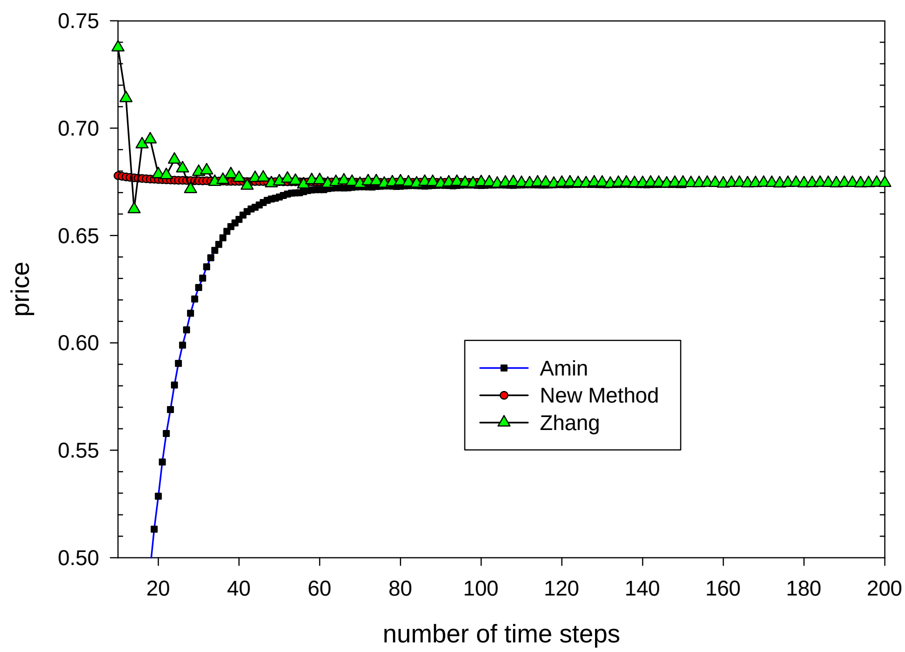


Figure B.16: Prices varying the number of time steps [Case IIA]

Parameters: S=40, X=30, r=8%, $\sigma^2 = 0.05$, $\delta^2 = 0.05$, $\lambda = 5.0$, k = 0, maturity=0.25 year.

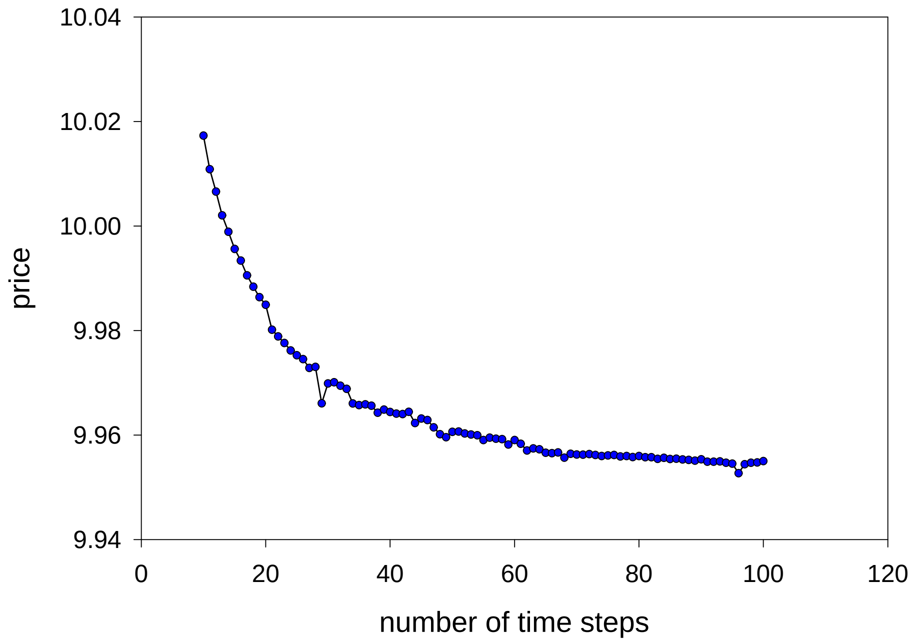


Figure B.17: Prices varying the number of time steps using Ext. integral eqn method [Case IIB]

Parameters: S=40, X=45, r=8%, $\sigma^2 = 0.05$, $\delta^2 = 0.05$, $\lambda = 5.0$, k = 0, maturity=1.0 year.

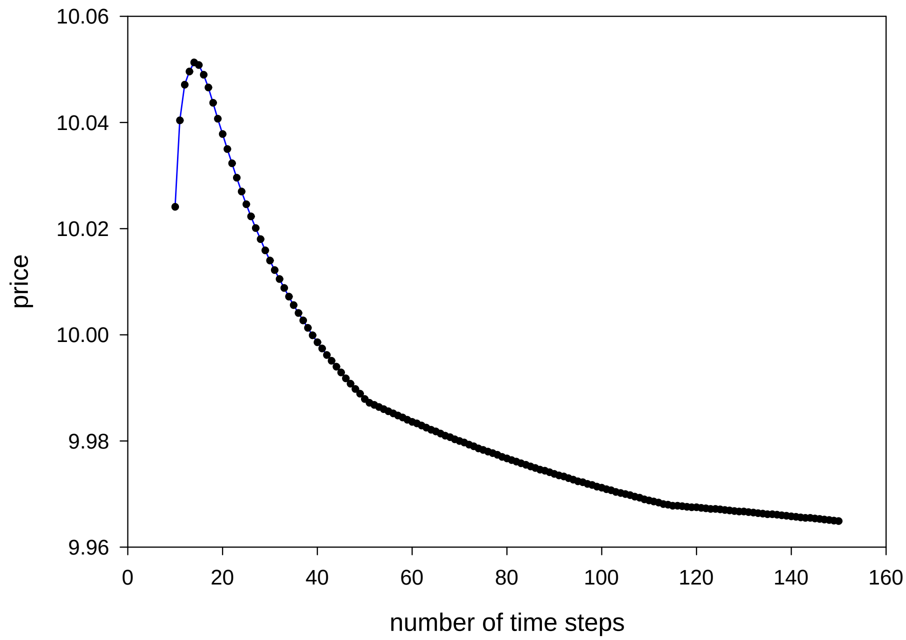


Figure B.18: Prices varying the number of time steps using Amin's method [Case IIB]

Parameters: S=40, X=45, r=8%, $\sigma^2 = 0.05$, $\delta^2 = 0.05$, $\lambda = 5.0$, k = 0, maturity=1.0 year.

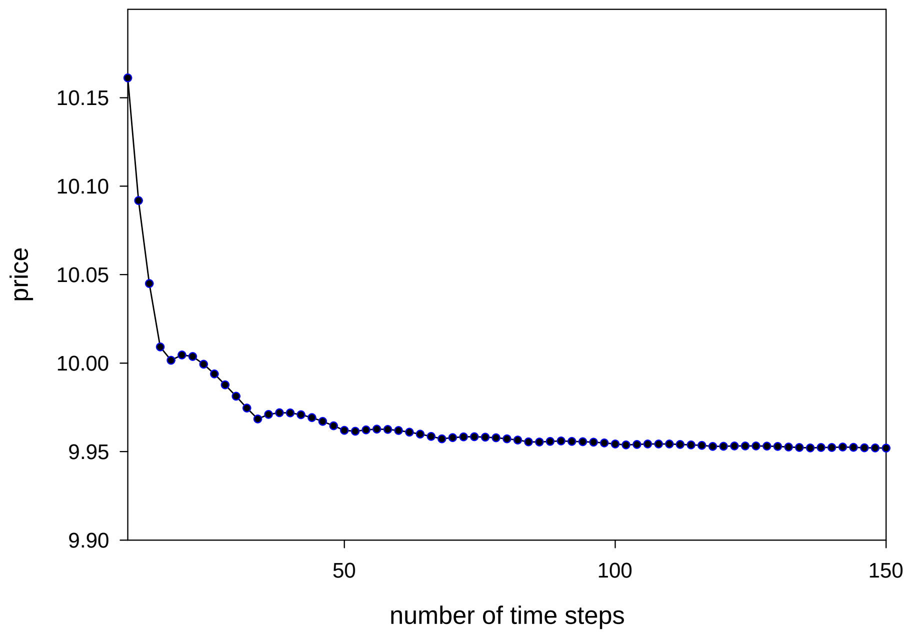


Figure B.19: Prices varying the number of time steps using Zhang's method [Case IIB]

Parameters: S=40, X=45, r=8%, $\sigma^2 = 0.05$, $\delta^2 = 0.05$, $\lambda = 5.0$, k = 0, maturity=1.0 year.

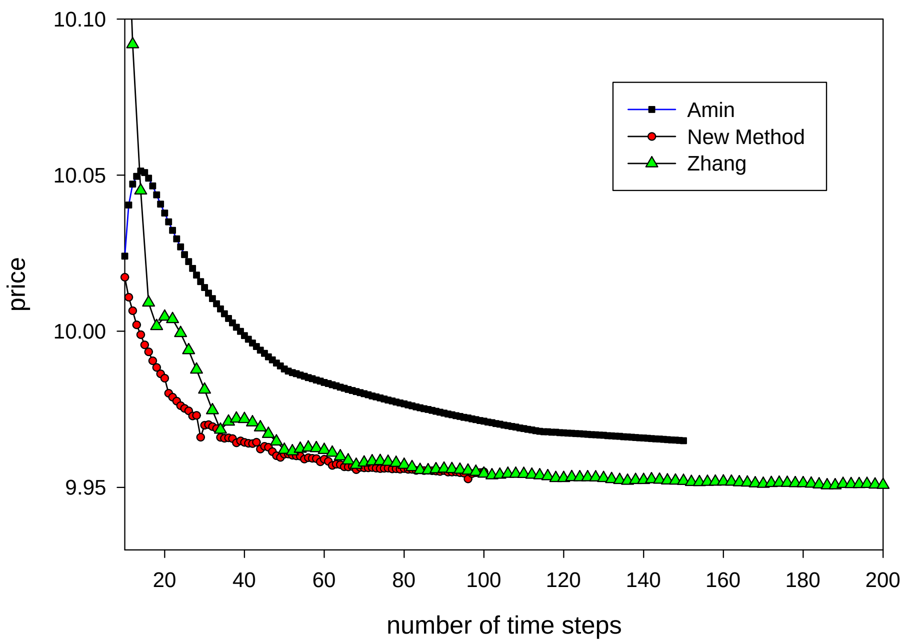


Figure B.20: Prices varying the number of time steps [Case IIB]

Parameters: S=40, X=45, r=8%, $\sigma^2 = 0.05$, $\delta^2 = 0.05$, $\lambda = 5.0$, k = 0, maturity=1.0 year.

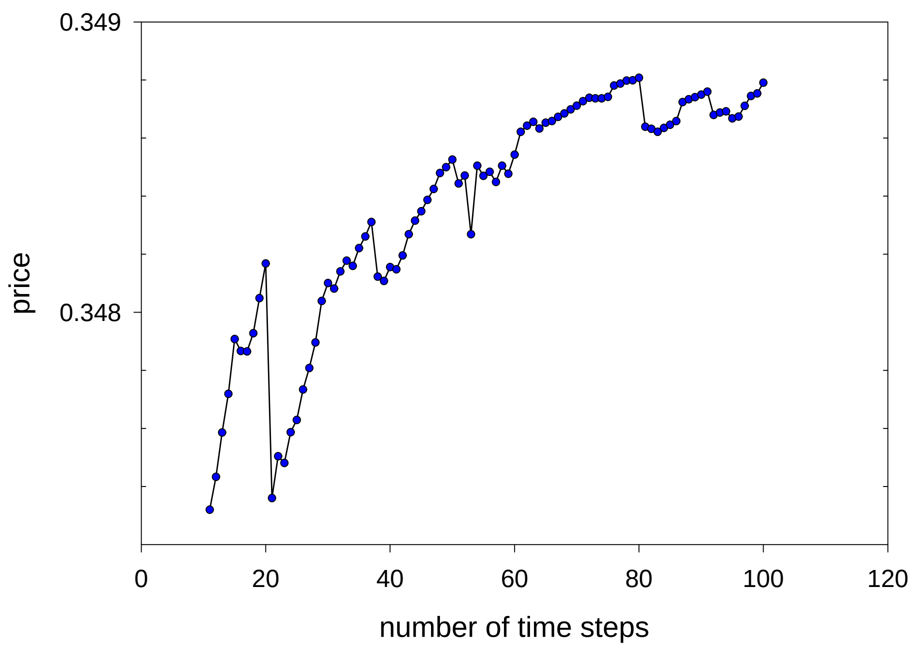


Figure B.21: Prices varying the number of time steps using Ext. integral eqn method [Case IIIA]

Parameters: S=50, X=45, r=9%, σ^2 = 0.004, δ^2 = 0.039220713, λ = 0.9, k = 0, maturity=0.25 year.

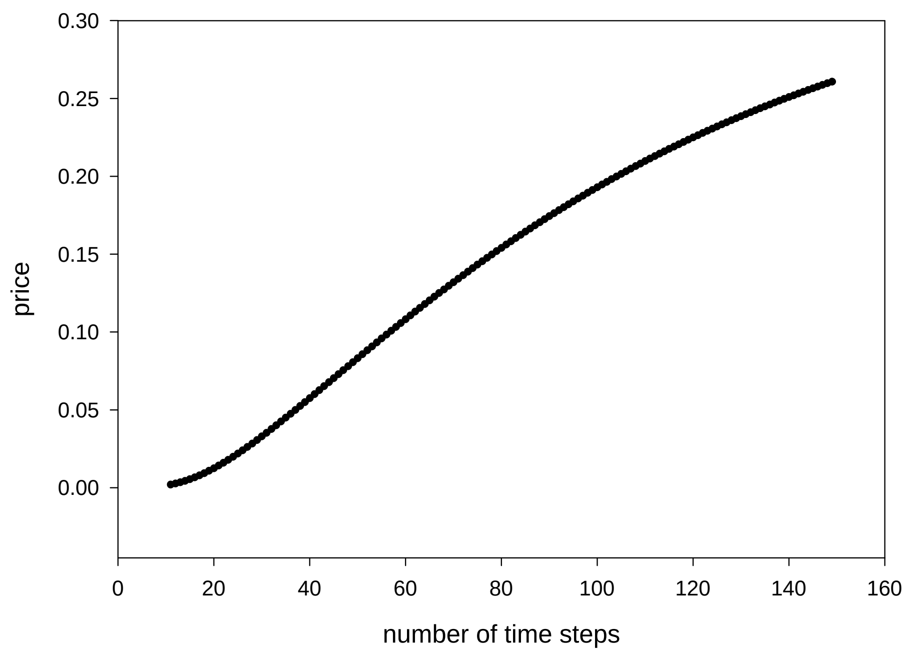


Figure B.22: Prices varying the number of time steps using Amin's method [Case IIIA]

Parameters: S=50, X=45, r=9%, $\sigma^2 = 0.004$, $\delta^2 = 0.039220713$, $\lambda = 0.9$, k = 0, maturity=0.25 year.

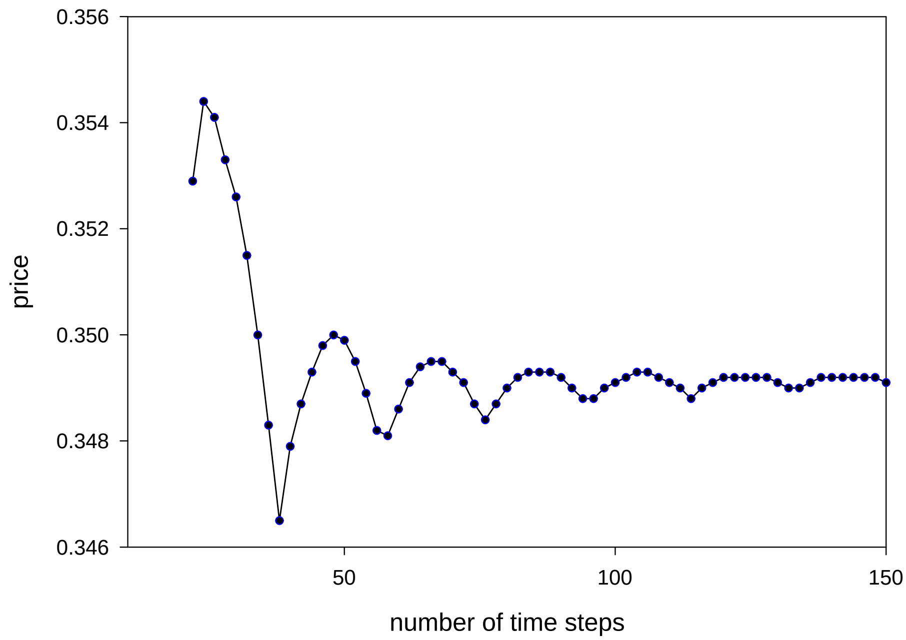


Figure B.23: Prices varying the number of time steps using Zhang's method [Case IIIA]

Parameters: S=50, X=45, r=9%, $\sigma^2 = 0.004$, $\delta^2 = 0.039220713$, $\lambda = 0.9$, k = 0, maturity=0.25 year.

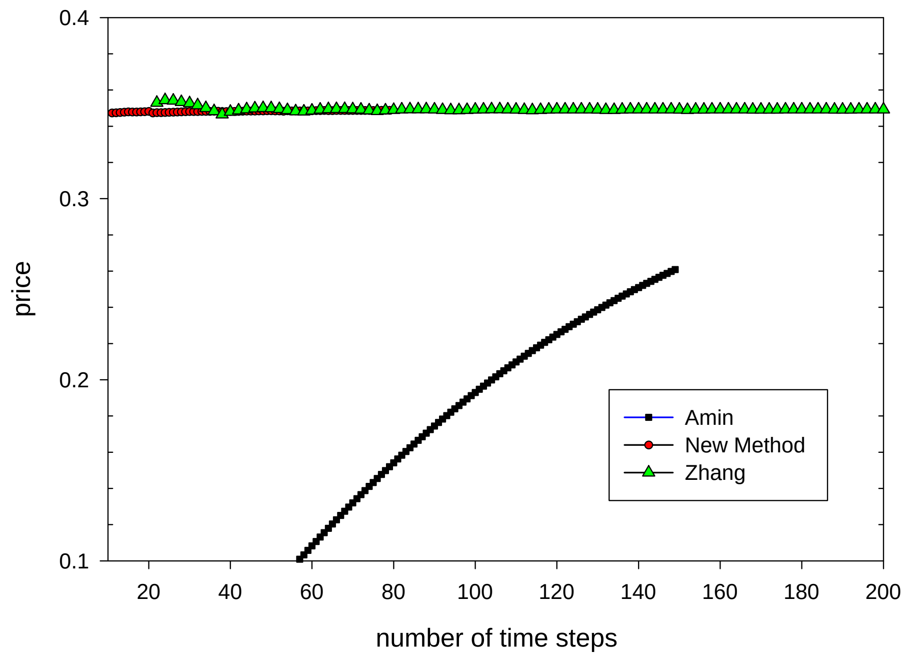


Figure B.24: Prices varying the number of time steps [Case IIIA]

Parameters: S=50, X=45, r=9%, $\sigma^2 = 0.004$, $\delta^2 = 0.039220713$, $\lambda = 0.9$, k = 0, maturity=0.25 year.

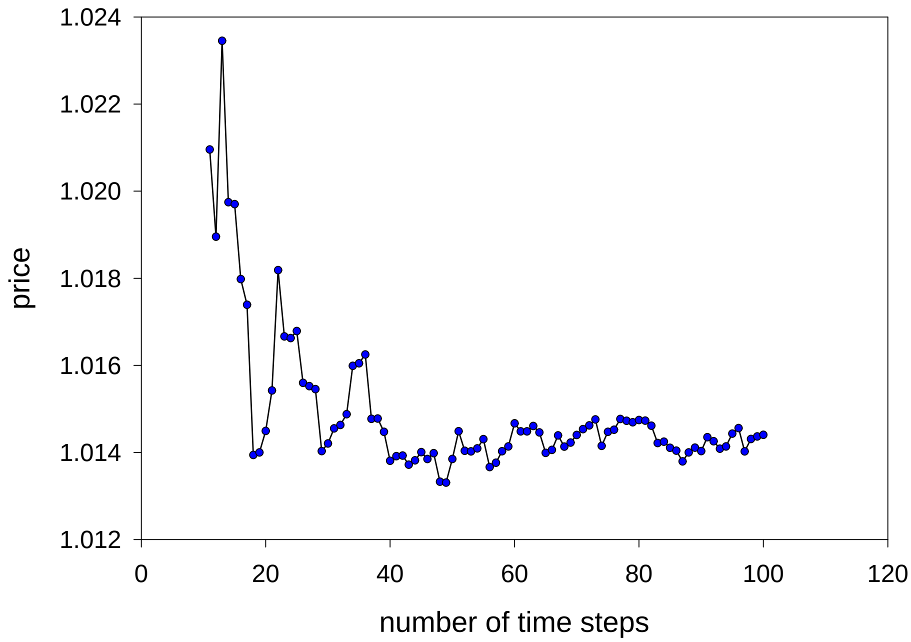


Figure B.25: Prices varying the number of time steps using Ext. integral eqn method [Case IIIB]

Parameters: S=50, X=45, r=9%, $\sigma^2 = 0.004$, $\delta^2 = 0.039220713$, $\lambda = 0.9$, k = 0, maturity=1.0 year.

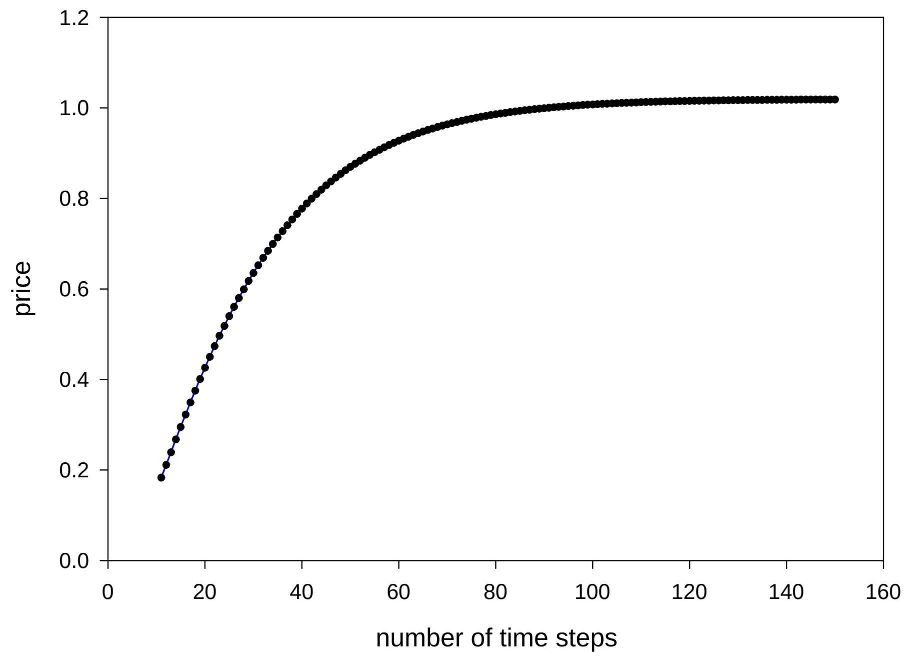


Figure B.26: Prices varying the number of time steps using Amin's method [Case IIIB]

Parameters: S=50, X=45, r=9%, $\sigma^2 = 0.004$, $\delta^2 = 0.039220713$, $\lambda = 0.9$, k = 0, maturity=1.0 year.

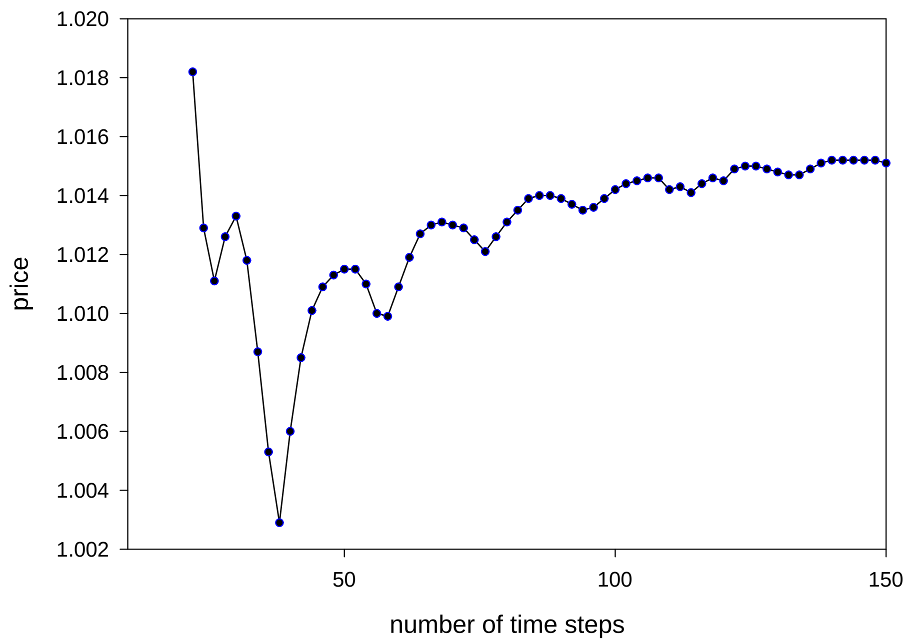


Figure B.27: Prices varying the number of time steps using Zhang's method [Case IIIB]

Parameters: S=50, X=45, r=9%, σ^2 = 0.004, δ^2 = 0.039220713, λ = 0.9, k = 0, maturity=1.0 year.

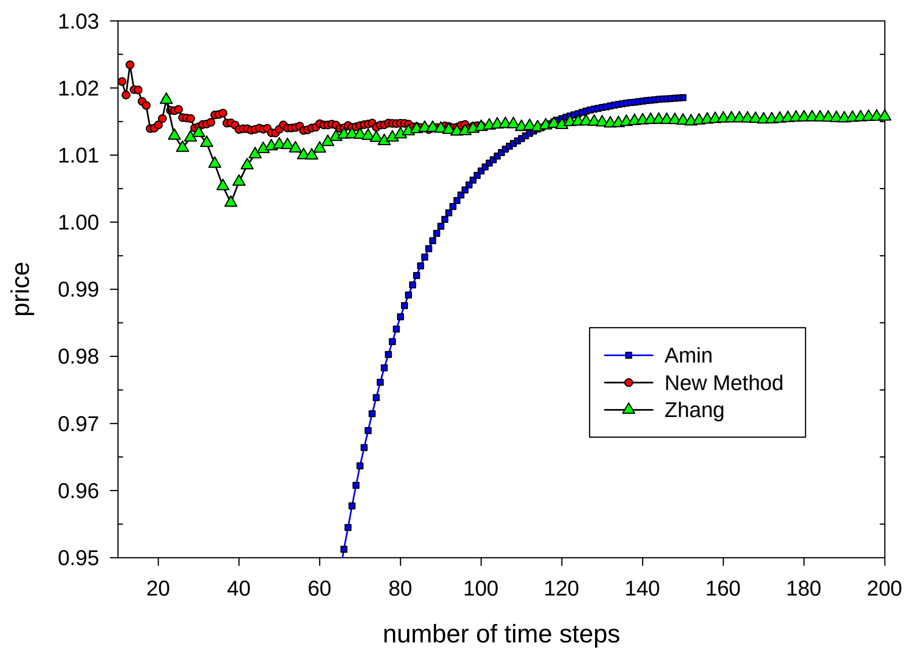
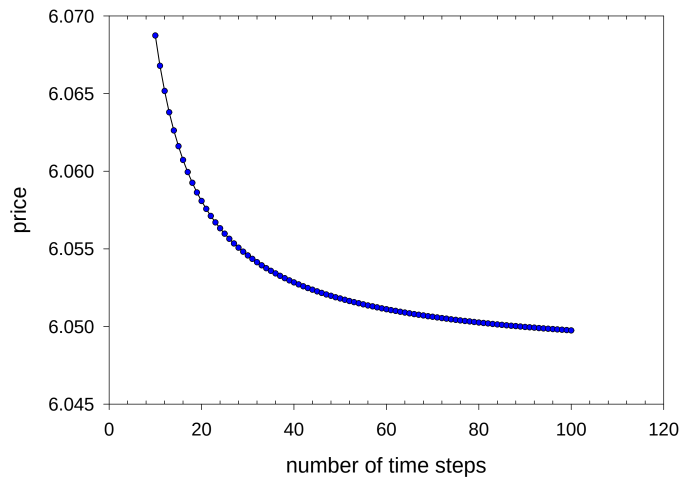


Figure B.28: Prices varying the number of time steps [Case IIIB]

Parameters: S=50, X=45, r=9%, $\sigma^2 = 0.004$, $\delta^2 = 0.039220713$, $\lambda = 0.9$, k = 0, maturity=1.0 year.



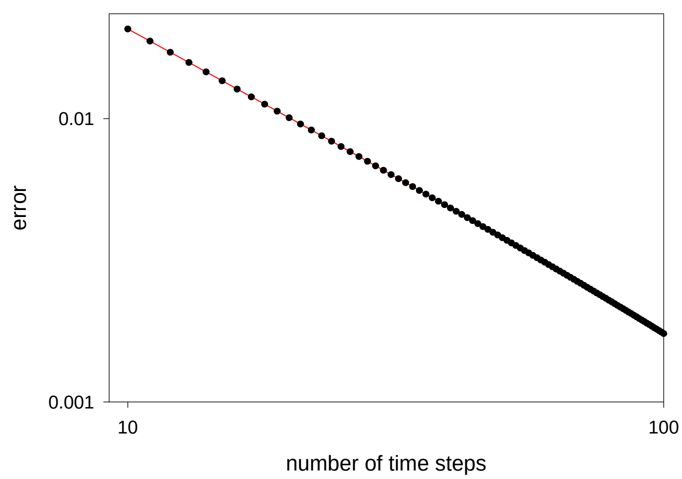
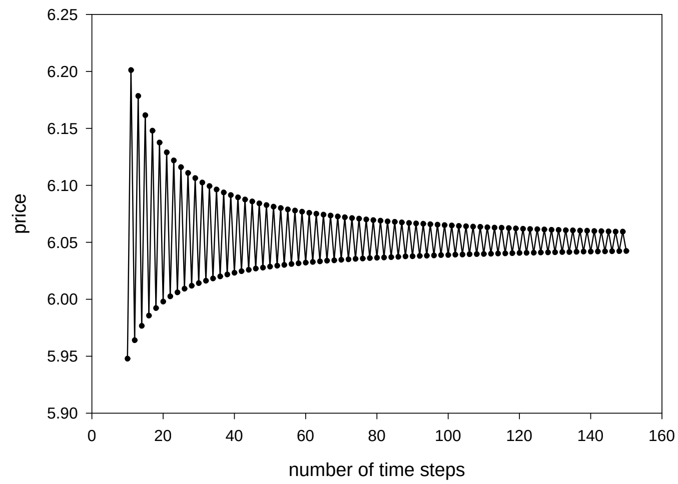


Figure B.29: Prices and errors varying the number of time steps using Ext. integral eqn method [Case IC]

Parameters: S=100, X=100, r=6%, $\sigma^2 = 0.1127577$, $\delta = 0.15$, $\lambda = 0.0001, k = 0$, maturity=0.25 year. The error e_N is defined as $|p_\infty - p_N|$.



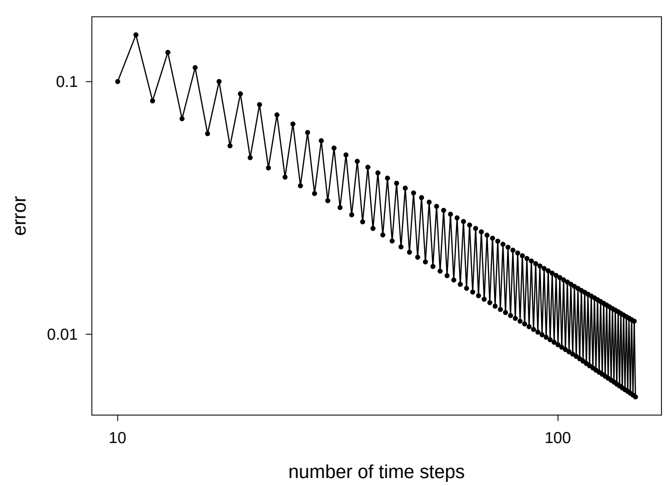
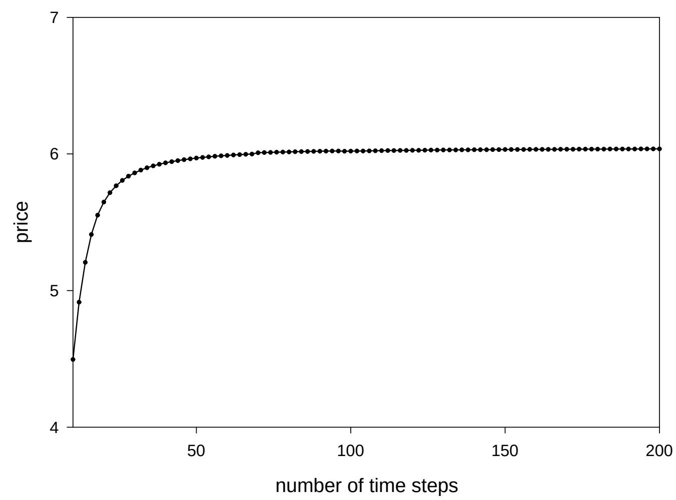


Figure B.30: Prices and errors varying the number of time steps using Amin's method [Case IC]

Parameters: S=100, X=100, r=6%, $\sigma^2 = 0.1127577$, $\delta = 0.15$, $\lambda = 0.0001, k = 0$, maturity=0.25 year. The error e_N is defined as $|p_\infty - p_N|$.



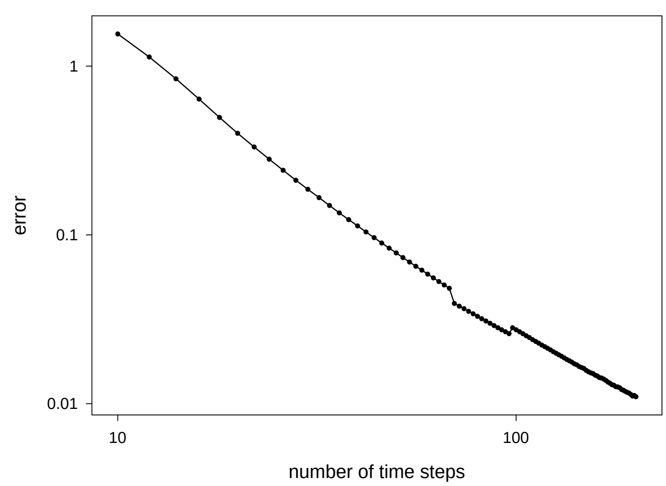
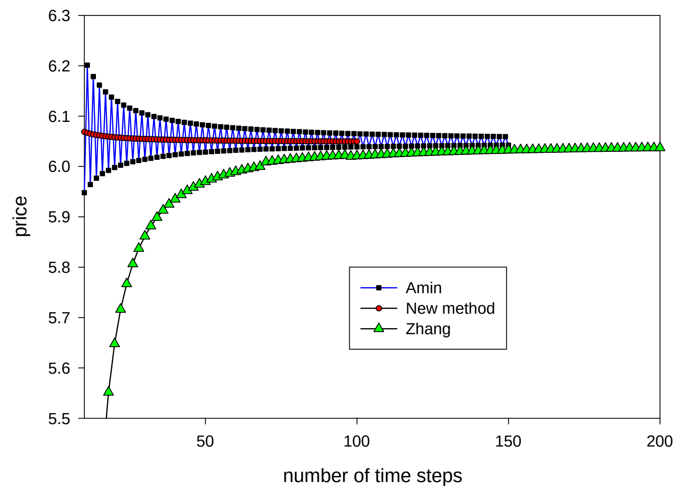


Figure B.31: Prices and errors varying the number of time steps using Zhang's method [Case IC]

Parameters: S=100, X=100, r=6%, $\sigma^2 = 0.1127577$, $\delta = 0.15$, $\lambda = 0.0001, k = 0$, maturity=0.25 year. The error e_N is defined as $|p_\infty - p_N|$.



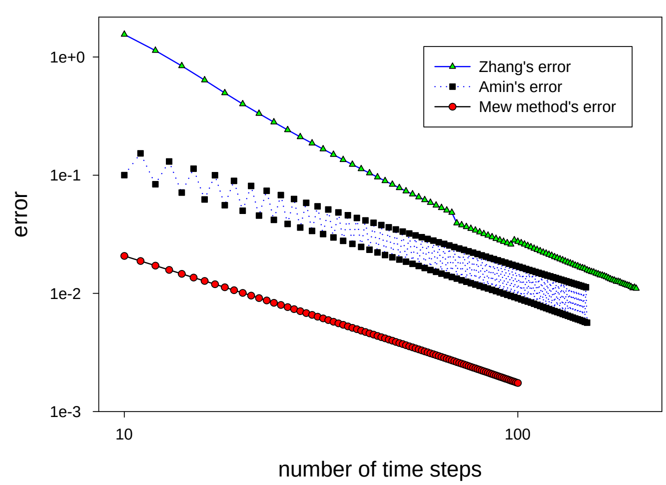
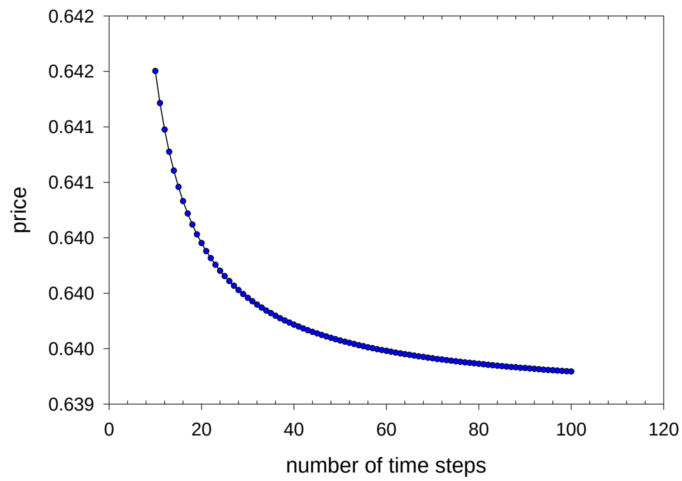


Figure B.32: Prices and errors varying the number of time steps using Zhang's method [Case IC]

Parameters: S=100, X=100, r=6%, $\sigma^2 = 0.1127577$, $\delta = 0.15$, $\lambda = 0.0001, k = 0$, maturity=0.25 year. The error e_N is defined as $|p_\infty - p_N|$.



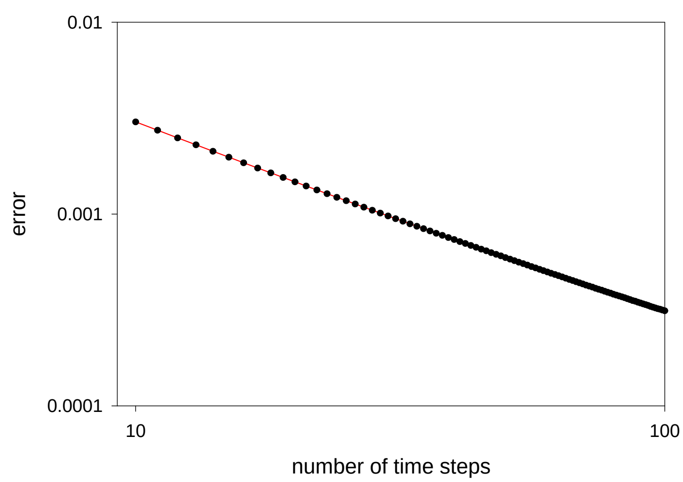
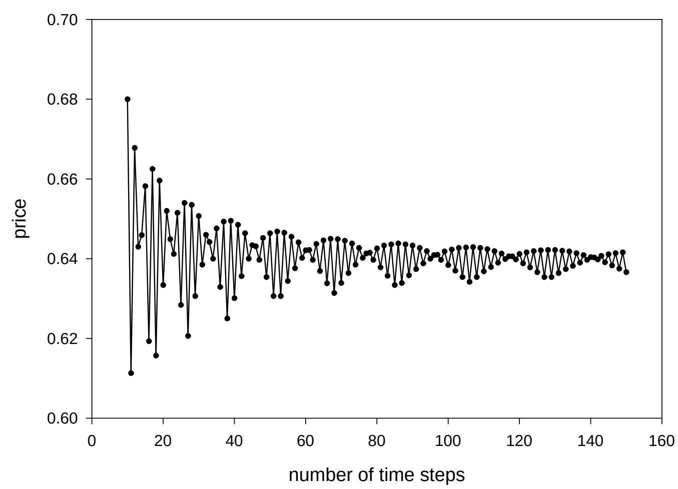


Figure B.33: Prices and errors varying the number of time steps using Ext. integral eqn method [Case IIC]

Parameters: S=40, X=30, r=8%, $\sigma^2 = 0.30635035$, $\delta^2 = 0.05$, $\lambda = 0.0001, k = 0$, maturity=0.25 year. The error e_N is defined as $|p_\infty - p_N|$.



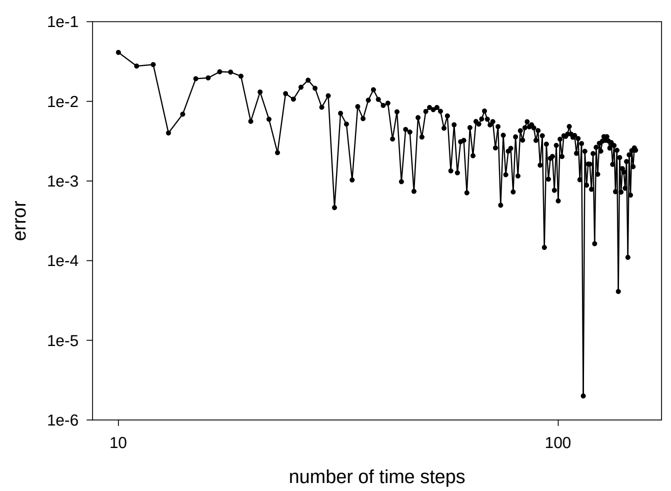
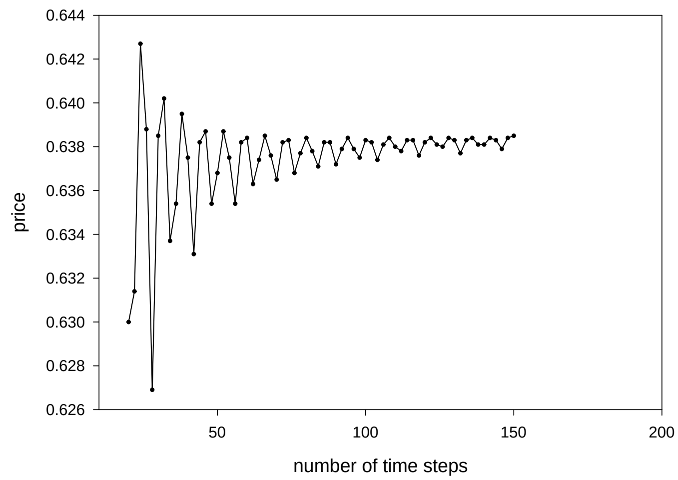


Figure B.34: Prices and errors varying the number of time steps using Amin's method [Case IIC]

Parameters: S=40, X=30, r=8%, $\sigma^2 = 0.30635035$, $\delta^2 = 0.05$, $\lambda = 0.0001, k = 0$, maturity=0.25 year. The error e_N is defined as $|p_\infty - p_N|$.



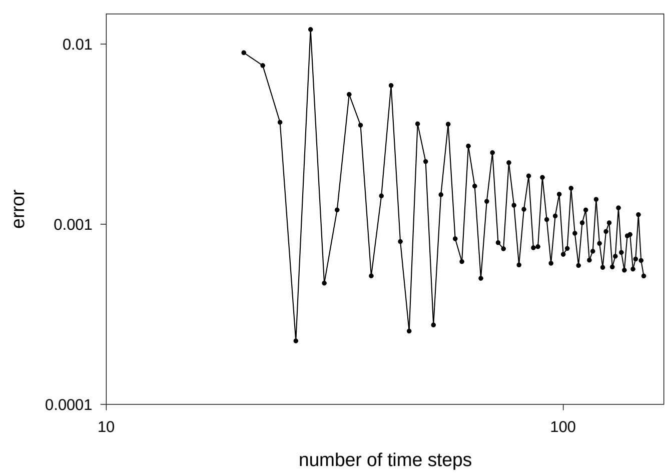
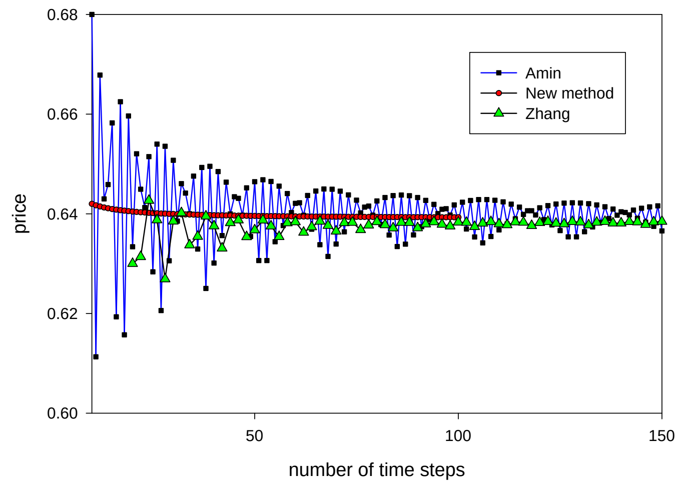


Figure B.35: Prices and errors varying the number of time steps using Zhang's method [Case IIC]

Parameters: S=40, X=30, r=8%, $\sigma^2 = 0.30635035$, $\delta^2 = 0.05$, $\lambda = 0.0001, k = 0$, maturity=0.25 year. The error e_N is defined as $|p_\infty - p_N|$.



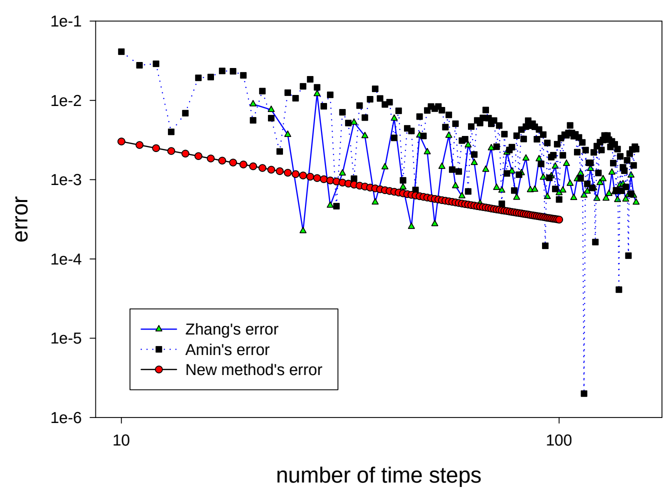
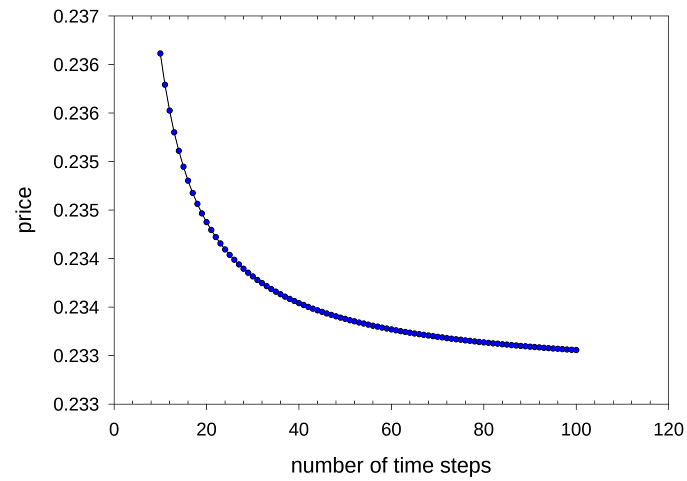


Figure B.36: Prices and errors varying the number of time steps using Zhang's method [Case IIC]

Parameters: S=40, X=30, r=8%, $\sigma^2 = 0.30635035$, $\delta^2 = 0.05$, $\lambda = 0.0001, k = 0$, maturity=0.25 year. The error e_N is defined as $|p_\infty - p_N|$.



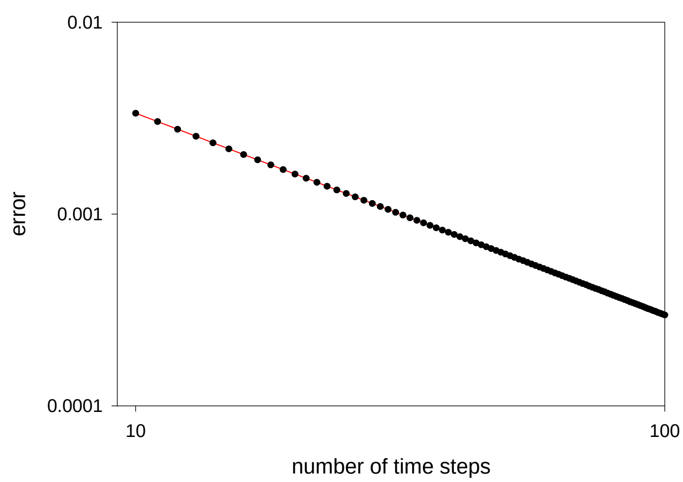
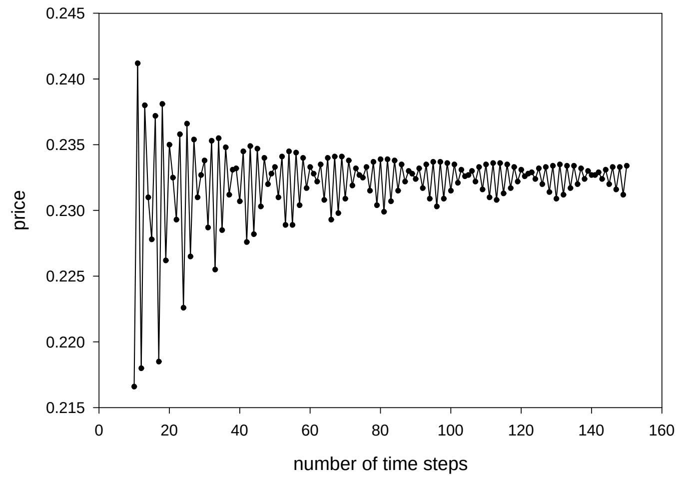


Figure B.37: Prices and errors varying the number of time steps using Ext. integral eqn method [Case IIIC]

Parameters: S=50, X=45, r=9%, $\sigma^2 = 0.03996$, $\delta^2 = 0.039220713$, $\lambda = 0.0001, k = 0$, maturity=0.25 year. The error e_N is defined as $|p_\infty - p_N|$.



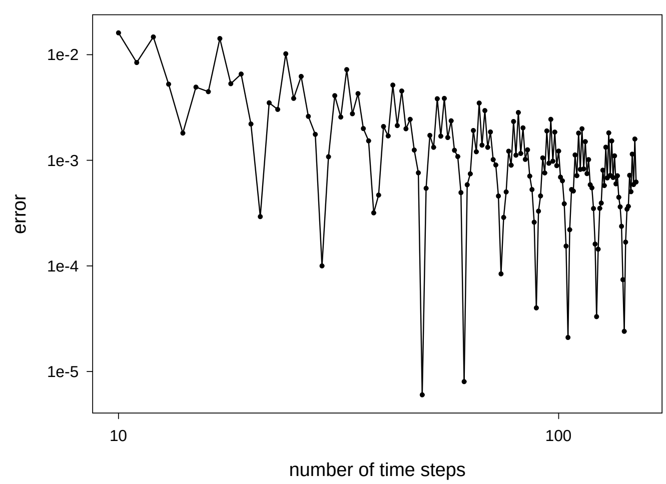
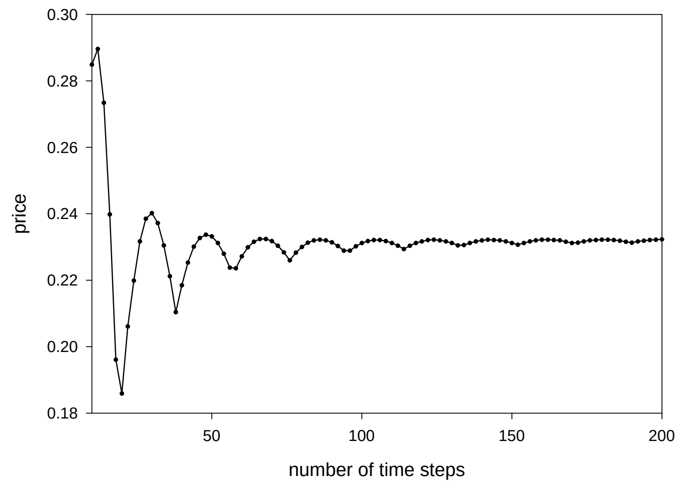


Figure B.38: Prices and errors varying the number of time steps using Amin's method [Case IIIC]

Parameters: S=50, X=45, r=9%, $\sigma^2 = 0.03996$, $\delta^2 = 0.039220713$, $\lambda = 0.0001, k = 0$, maturity=0.25 year. The error e_N is defined as $|p_\infty - p_N|$.



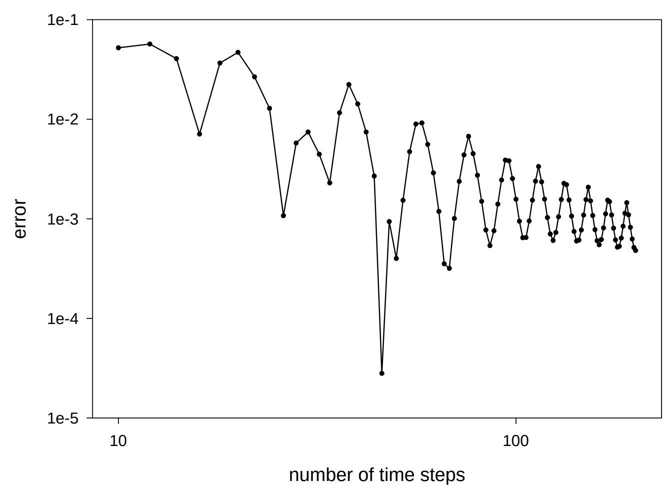
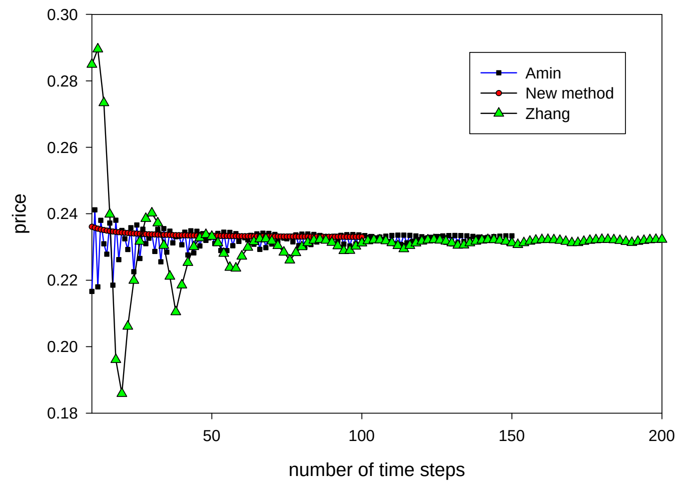


Figure B.39: Prices and errors varying the number of time steps using Zhang's method [Case IIIC]

Parameters: S=50, X=45, r=9%, $\sigma^2 = 0.03996$, $\delta^2 = 0.039220713$, $\lambda = 0.0001, k = 0$, maturity=0.25 year. The error e_N is defined as $|p_\infty - p_N|$.



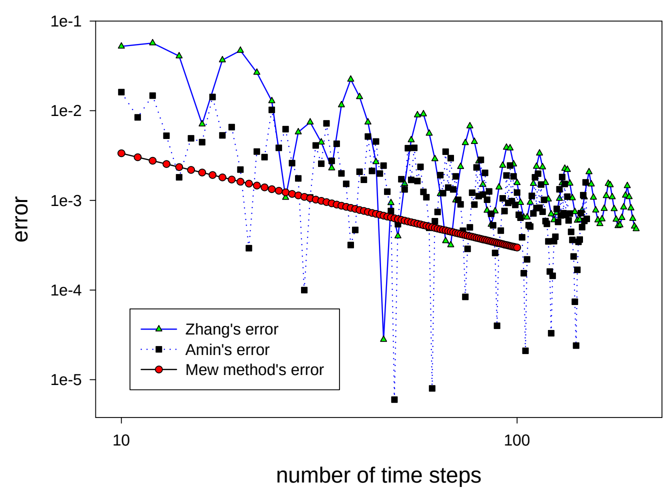


Figure B.40: Prices and errors varying the number of time steps using Zhang's method [Case IIIC]

Parameters: S=50, X=45, r=9%, $\sigma^2 = 0.03996$, $\delta^2 = 0.039220713$, $\lambda = 0.0001, k = 0$, maturity=0.25 year. The error e_N is defined as $|p_\infty - p_N|$.

Appendix C

C Codes

This Appendix presents computer code written in C language for the numerical solutions of various models for American option prices under jump-diffusion processes. The algorithms of the numerical methods are explained in Chapter 5. The code is merely for computing and comparing the numerical results in the Appendix B, and is not intended for commercial or other uses. Although this code has been successfuly applied to various examples as shown in the previous Appendix, the author does not accept any liability for any losses related to its use, or misuse, for any purpose.

The code should be linked with the following copyrighted subroutines from *Numerical Recipes in C, The Art of Scientific Computing*, by Press, Flannery, Teukolsky, and Vetterling (1997).

- Extended integral equation method: `bccbug()`, `zbrak()`, `rtsafe()`, `nrutil()`
- Amin's method: No linked subroutine needed.
- Zhang's method: `tridag()`, `nrutil()`, `bccbug()`
- MacMillan-Zhang's method: `nrutil()`, `bccbug()`, `zbrak()`, `rtsafe()`

- Modified MacMillan-Zhang's method:`nrutil()`, `bccbug()`, `zbrak()`, `rtsafe()`
- Extended Ju's method: `nrutil()`, `broyden()`, `newt()`, `fmin()`, `lubksb()`, `ludcmp()`, `fdjac()`, `qrdcmp()`, `rsolv()`, `rotate()`, `rtsafe()`, `zbrak()`

Also most programs need the header files, `nr.h`, `nrutil.h`, to successfuly link the subroutines above.

C.1 Extended integral equation method

```
/*************************************************/
/*     Calculation of an American put price      */
/*        in the jump-diffusion model            */
/*        using the integral equation method     */
/*     Written by Byeong-wook Choi               */
/*        Created in June 2000                   */
/*        Lastly Modified in May 2002            */
/*************************************************/
#include <stdio.h>
#include <math.h>
#include <time.h>
#include <sys/types.h>
#include <unistd.h>
#include "nr.h"
#include "nrutil.h"
#define PSMAX     20
#define N        500
#define STEP     100
#define dt       MATURITY/STEP
/* Parameters of my example */
#define MATURITY 1.0
#define r        0.06
#define S0       80
#define X        100
#define sig      0.33564
```

```
#define lambda   1.0
#define kappa    0.0
#define jsig     0.01
/* For Newton-Raphson Method */
#define NN    100
#define NBMAX    20
float efunc(float x);
float efunc1(float x);
/** Global variables **/
int Stime;
double crit_price[N];
/** Computing Option Prices  **/
float jprob(float,float, float);
float joption(float, float);

static float fx(float x)
{
    return efunc(x);
}
static void funcd(float x,float *fn, float *df)
{
    *fn = efunc (x);
    *df = efunc1(x);
}
main()
{
    int i,j,k;
    double X1, X2, price_1, sum_price_n;
    /* For Newton-Raphson Method */
    int nb=NBMAX;
    float xacc,root,*xb1,*xb2;
    crit_price[0] =X;
    /* Finding a root */
    for(Stime=1; Stime<=STEP; Stime++) {
        X1=1.0;        /* Should be greater than 0 */
        X2=X;
        xb1=vector(1,NBMAX);
        xb2=vector(1,NBMAX);
        zbrak(fx,X1,X2,NN,xb1,xb2,&nb);
```

```
        for (i=1;i<=nb;i++) {
            xacc=(1.0e-6)*(xb1[i]+xb2[i])/2.0;
            root=rtsafe(funcd,xb1[i],xb2[i],xacc);
        }
        crit_price[Stime] = root;
        free_vector(xb2,1,NBMAX);
        free_vector(xb1,1,NBMAX);
    }
    /* Computing an American option price */
    price_1 = joption(S_0, MATURITY);
    sum_price_n = price_1
         + 0.5*dt*r*X*exp(-r*MATURITY)*jprob(S_0, X, MATURITY);
    for(k=1; k<STEP; k++)
        sum_price_n += (r*X*MATURITY/STEP) *
                   ( exp(-r*(STEP-k)*dt)
                 *jprob(S_0, crit_price[k], (STEP-k)*dt));
    printf("Option price is equal to %f\n", sum_price_n);
}
double fact(int n) /* MAX VALUE=171 */
{
    if(n == 0) return(1);
    else return(n*fact(n-1));
}
float joption(float S, float maturity)
{
    float noption(float, float, float, float);
    double poisson_val(float, int);
    double jump_sig, jump_r, sum_value;
    int i;
    sum_value = 0;
    for(i=0; i< PSMAX; i++) {
        jump_r = r - lambda*kappa + i*log(1+kappa)/maturity;
        jump_sig = sqrt(pow(sig,2)+i*pow(jsig,2)/maturity);
        sum_value += poisson_val(lambda*(1+kappa)*maturity,i)
                *noption(S, maturity, jump_r, jump_sig);
    }
    return(sum_value);
}
double joption_d(double S, double maturity)
```

```
{
    double fact(int);
    float nprob(float);
    double poisson_val(float, int);
    double jump_sig, jump_r, sum_value, dd1;
    int i;
    sum_value = 0;
    for(i=0; i< PSMAX; i++) {
        jump_r = r - lambda*kappa + i*log(1+kappa)/maturity;
        jump_sig = sqrt(pow(sig,2)+i*pow(jsig,2)/maturity);
        dd1 = (log(S)-log(X)+(jump_r+pow(jump_sig,2)/2.)*maturity)
            / (jump_sig*sqrt(maturity));
        sum_value += poisson_val(lambda*(1+kappa)*maturity,i)
                *(-nprob(-dd1));
    }
    return(sum_value);
}
float jprob(float S, float K, float mat)
{
    float nprob(float);
    double poisson_val(float, int);
    double fact(int);
    double jump_r, jump_sig, dd2, sum;
    int i;
    sum=0;
    for(i=0; i<PSMAX; i++) {
        jump_r = r-lambda*kappa + i*log(1+kappa)/mat;
        jump_sig = sqrt(pow(sig,2)+i*pow(jsig,2)/mat);
        dd2 = (log(S)-log(K)+(jump_r-0.5*pow(jump_sig,2))*mat)
            /(jump_sig*sqrt(mat));

        sum += poisson_val(lambda*mat,i)*nprob(-dd2);
    }
    return(sum);
}
double jprob_d(double S, double K, double maturity)
{
    double fact(int);
    double poisson_val(float, int);
```

```
    double sum_value, jump_sig, jump_r, dd2;
    int i;
    sum_value = 0;
    for(i=0; i< PSMAX; i++) {
        jump_r = r - lambda*kappa + i*log(1+kappa)/maturity;
        jump_sig = sqrt(pow(sig,2)+i*pow(jsig,2)/maturity);
        dd2 = (log(S)-log(K)+(jump_r-pow(jump_sig,2)/2.)*maturity)
            / (jump_sig*sqrt(maturity));
        sum_value += poisson_val(lambda*maturity,i)
                *(1/sqrt(2*3.14159)*exp(-0.5*pow(dd2,2)))
                    *(-1/S/jump_sig/sqrt(maturity) );
    }
    return(sum_value);
}
float noption(float S0, float maturity, float rate, float volatility)
{
    float nprob(float);
    double dd1, dd2;
    double value;

    dd1 = ( log(S0)-log(X) + (rate+pow(volatility,2)/2.)*maturity )
        / (volatility*sqrt(maturity));
    dd2 = dd1 - volatility*sqrt(maturity);

    value = X*exp(-rate*maturity)*nprob(-dd2) -S0*nprob(-dd1);

    return(value);
}
/* Compute Poisson value */
double poisson_val(float coeff, int n)
{
    int i;
    double sumlog;
    if (n==0) return exp(-coeff);
    sumlog=0;
    for (i=1; i<=n; i++)
        sumlog += log(i);
    return exp(-coeff + n*log(coeff) - sumlog);
}
```

```
/* External Variables */
extern int Stime;
extern double crit_price[];
float efunc(float y)
{
    float jprob(float, float, float);
    float joption(float, float);
    double ans, option_value, another_value, sum_prob;
    int j;
    option_value = joption(y, Stime*dt);
    another_value = 0.5*r*X*exp(-Stime*r*dt)*dt*jprob(y,X,Stime*dt);
    sum_prob = 0;
    for(j=1; j<Stime; j++)
        sum_prob += r*X*exp(-j*r*dt)*dt
                *jprob(y,crit_price[Stime-j],j*dt);
    ans = option_value + another_value + sum_prob+y-X;
    return ans;
}

/* External variables */
extern double crit_price[];
extern int Stime;
float efunc1(float y)
{
    float jprob_d(float,float,float);
    float joption_d(float,float);
    double ans, sum_prob1, dd2;
    double option_value, another_value;
    float d1; int j;
    option_value = joption_d(y, Stime*dt) ;
    another_value = 0.5*r*X*exp(-Stime*r*dt)*dt*jprob_d(y,X,Stime*dt);
    sum_prob1 = 0;
    for(j=1; j<Stime; j++)
        sum_prob1 += r*X*exp(-j*r*dt)*dt
                *jprob_d(y, crit_price[Stime-j],j*dt);
    ans = option_value + another_value + sum_prob1 + 1;
    return ans;
}
```

C.2 Amin's method

```
/***********************************************/
/*      This program is for the calculation of    */
/*             American Put prices                */
/*             based on Amin's method (JF 1993)   */
/*                                                */
/*      Written by Byeong-wook Choi               */
/*             August 13, 2000                    */
/***********************************************/
#include <stdio.h>
#include <math.h>
#include <sys/types.h>
#include <unistd.h>

#define EURO_CALL   1
#define AMER_CALL   2
#define EURO_PUT    3
#define AMER_PUT    4
#define MAX(x,y) (x>=y)? x:y
#define Min(x,y) (x>=y)? y:x
#define N        200050  /* Max duration: > NSTEPS*LL*2 */
#define M        200050  /* Max duration: > 2*LL */
/* We don't use NSTEPS, LL */
/* Number of Steps (numstep) <= 200 */
/* Number of jumps (lstep) <= 200 */
double amin(int type, double S0, double X, double r,
        double sig, double delta, double LAMBDA,
        double maturity, int nstep, int jumpstep, double jump_size)
{
    double nprob(double);
    double stockprice(int, int, double, double, double, double, int);
    double jump_mean;
    double *payoff= (double *) malloc(sizeof(double)*N);
    double *call = (double *) malloc(sizeof(double)*N);
    double *new_call = (double *) malloc(sizeof(double)*N);
    double *Q = (double *) malloc(sizeof(double)*M);
    double *dN_n = (double *) malloc(sizeof(double)*M);
    double weight, qsum, q_n, dt, discount, alpha, w, up;
    int L;
```

```
    register int i, j, k, kk, tt;
    /******************/
    /* Jump parameter */
    /******************/
    dt = maturity/nstep;
    discount = exp(-r*dt);
    alpha = r - 0.5*pow(sig,2) - LAMBDA*jump_size;
    w = exp(alpha*dt);
    up = exp(sig*sqrt(dt));
    /* Mean and variance of the jump distribution !! */
    jump_mean = -0.5*delta*delta;
    q_n = 0.5;
    L=jumpstep;  /* No truncation */
    /* Truncation method */
    /***
    for(k=1;k <jumpstep+1; k++)
        if (alpha*dt-k*sig*sqrt(dt) <= -3*delta
            && alpha*dt+k*sig*sqrt(dt) >= 3*delta) {
            L = k;
            break;
        }
    ***/
    /****************************/
    /* Transition Probabilities  */
    /****************************/
    for(k=0; k <2*L+1; k++) {
        dN_n[k] =0;
        Q[k] = 0;
    }
    dN_n[0] =
nprob((alpha*dt + (1+0.5)*sig*sqrt(dt)-jump_mean)/delta)
        - nprob((alpha*dt + (-1-0.5)*sig*sqrt(dt)-jump_mean)/delta);
    for(k=1; k <2*L+1; k++) {
        if (k==1 || k==L+1) {
            dN_n[k] = 0;
        }
        else if (k <= L) {
            dN_n[k] =
      nprob((alpha*dt + (k+0.5)*sig*sqrt(dt)-jump_mean)/delta)
```

```
          - nprob((alpha*dt + (k-0.5)*sig*sqrt(dt)-jump_mean)/delta);
    }
    else {
        kk = -(k-L);
        dN_n[k] =
  nprob((alpha*dt + (kk+0.5)*sig*sqrt(dt)-jump_mean)/delta)
          - nprob((alpha*dt + (kk-0.5)*sig*sqrt(dt)-jump_mean)/delta);
    }
}
for(k=0; k <2*L+1; k++)
    Q[k] = LAMBDA*dt*dN_n[k];
Q[1]     = q_n * (1-LAMBDA*dt);
Q[L+1]   = (1-q_n) * (1-LAMBDA*dt);
qsum=0;
for(k=0; k <2*L+1; k++) {
    qsum += Q[k];
}
/********************************/
/* Initialization  */
/********************************/
for(j=0; j<2*nstep*L+1 ; j++) {
    call[j] = MAX(X-stockprice(nstep, j, w, up, S0, X, L), 0);
    payoff[j] =0;
    new_call[j] =0;
}
/********************************/
/* Backward Dynamic Programming */
/********************************/
for(i=nstep-1; i>=0; i--) {
    /*--------------------------------------*/
    /* Calculation of temporary option price */
    /*   for American option valuation      */
    /*--------------------------------------*/
    for(j=0; j<2*i*L+1; j++) {
        payoff[j] = MAX(X-stockprice(i, j, w, up, S0,X,L), 0);
    }
    for(j=0; j<=i*L; j++)  {
        weight = 0;
        for (k=0; k<=L; k++)  {            /** SAME and UP **/
```

```
                weight +=  Q[k]*call[j+k];
            }
            for (k=1; k<=L; k++) {            /** DOWN **/
                tt = (j>=k)? (j-k) : (i+1)*L-(j-k);
                weight +=  Q[k+L]*call[tt];
            }
            new_call[j] = MAX(discount*weight, payoff[j]);
        }
        for(j=i*L+1; j<2*i*L+1; j++)  {
            weight =  Q[0]*call[j+L];        /** SAME **/
            for (k=1; k<=L; k++) {             /** DOWN **/
                weight +=  Q[k+L]*call[j+L+k];
            }
            for (k=1; k<=L; k++) {            /** UP **/
                tt = (j+L-k > (i+1)*L) ? (j+L-k) : (i+1)*L-(j+L-k);
                weight +=  Q[k]*call[tt];
            }
            new_call[j] = MAX(discount*weight, payoff[j]);
        }
        for(j=0; j<2*i*L+1; j++) {
            call[j] = new_call[j];
        }
    }
    return(call[0]);
}
/*** Stock Price Process ***/
double stockprice(int time, int state, double w,
                  double up, double S, double K, int bound)
{
    double stock_process;
    stock_process = 0;
    if (state <= time*bound)
        stock_process = S*pow(w,time)*pow(up,state);
    else if (state > time*bound)
        stock_process = S*pow(w,time)*pow(1/up,state-time*bound);
    return Min(K, stock_process); /* Add K for simple computation */
}
/*****************************************************
* Compute Normal cumulative distribution probability
```

```
*************************************************/
#define     PI      4.0e0*atan(1.e0)
#define     R       0.2316419
#define     A1      0.319381530
#define     A2      -0.356563782
#define     A3      1.781477937
#define     A4      -1.821255978
#define     A5      1.330274429
/* Ref. Hull's book p.243 */
/**************** NOTE ****************************
    N(x) = 1-N'(x)(a1k + a2k2 + a3k3 + a4k4 + a5k5)
             = 1-N(-x)
    k = 1/(1+rx)
    r = 0.2316419
    a1 = 0.319381530
    a2 = -0.356563782
    a3 = 1.781477937
    a4 = -1.821255978
    a5 = 1.330274429
    N'(x) = (1/sqrt(2pi))exp(-x^2/2)
**************************************************/
double nprob(double x)
{
    double K, Np, Norm;
    int flag;
    flag = 0;
    if (x>=0) flag = 1;
    else x=-x;
    K = 1.0/(1.0+R*x);
    Np = (1.0/sqrt(2*PI))*exp(-pow(x,2)/2);
    Norm = 1-Np*(A1*K+A2*pow(K,2)+A3*pow(K,3)+A4*pow(K,4)+A5*pow(K,5));
    return( (flag == 1)? Norm : 1-Norm );
}
```

C.3 Zhang's method

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
```

```
#include <sys/types.h>
#include <unistd.h>
#include "nr.h"
#include "nrutil.h"
#define MAX(a,b) (a,b, a>b ? a : b)
#define NP 1010     /* Maximal dimension of Tridiag */

float imfdm_lnj(float S0, float X, float r, float sig,
        float kappa, float jsig, float lambda,
        float T, int N, int M)
{
    float nprob(double);
    int i, j;
    unsigned long k,n;
    float *a,*b,*c,*rhs,*u, float **f;
    float jump1, jump2, prob1, prob2, probil, mean_j;
    double h, dS, dS0, RHS, LHS;
    int array_price;
    f=matrix(1,NP,1,NP);
    b=vector(1,NP);
    c=vector(1,NP);
    a=vector(1,NP);
    rhs=vector(1,NP);
    u=vector(1,NP);
    h   = T/N;
    dS  = 2./M;
    dS0 = log(S0)-1.0;;
    LHS = 1/h + r;
    RHS = (2*r-sig*sig)/(2*dS);
    a[1] = -0.5*h*pow(sig,2)/pow(dS,2)+0.5*(r-pow(sig,2)/2)*h/dS;
    b[1] = 1+h*pow(sig,2)/pow(dS,2) + r*h;
    c[1] = -0.5*h*pow(sig,2)/pow(dS,2)-0.5*(r-pow(sig,2)/2)*h/dS;
    for(j=2; j<=M-1; j++) {
        a[j] = a[1];
        b[j] = b[1];
        c[j] = c[1];
    }
    /* Payoff function at expiration date */
    for(j=1; j<=M+1; j++)
```

```
    f[N+1][j] = MAX(X-exp(dS0 + (j-1)*dS), 0);
/* Boundary condition */
for(i=1; i<=N+1; i++) {
    f[i][1] = MAX(X-exp(dS0),0);
    f[i][M+1] = 0;
}
mean_j = log(1+kappa)-0.5*jsig*jsig;
for(k=0; k<=N-1; k++) {
    /* Right hand side */
    for(i=1; i<M; i++)
        rhs[i] = f[N+1-k][1+i];
    /* Jump part */
    for(j=2; j<=M; j++) {
        if (jsig == 0) break;
        jump1=0;
        jump2=0;
        for(i=2; i<=M; i++) {
            prob1 = ((i-j+0.5)*dS-mean_j)/jsig;
            prob2 = ((i-j-0.5)*dS-mean_j)/jsig;
            probil = nprob(prob1)-nprob(prob2);
            jump1 += f[N+1-k][i]*probil;
        }
        for(i=j-1; i<=M; i++) {
            prob1 = ((-i+0.5)*dS-mean_j)/jsig;
            prob2 = ((-i-0.5)*dS-mean_j)/jsig;
            probil = nprob(prob1)-nprob(prob2);
            jump2 += MAX(X-exp(dS0
                + (-i+j-1)*dS),0)*probil;
        }
        for(i=1; i<=j-1; i++) {
            prob1 = ((-j+M+i+0.5)*dS-mean_j)/jsig;
            prob2 = ((-j+M+i-0.5)*dS-mean_j)/jsig;
            probil = nprob(prob1)-nprob(prob2);
            jump2 += MAX(X-exp(dS0
                + (M+i-1)*dS),0)*probil;
        }
        rhs[j-1] = f[N+1-k][j]
                   + lambda*h*(jump1+jump2-f[N+1-k][j]) ;
    }
```

```
        rhs[1] = rhs[1] - a[1]*f[N-k][1];
        rhs[M-1] = rhs[M-1] - c[1]*f[N-k][M+1];
        /* carry out solution */
        tridag(a,b,c,rhs,u,M-1);
        for(j=1; j<M; j++)
            f[N-k][1+j] = u[j];
        /* American option */
        for(j=1; j<M; j++)
            if(f[N-k][1+j] < X-exp(dS0+j*dS))
                f[N-k][1+j] = X-exp(dS0+j*dS);
    }
    array_price = M/2+1;
    free_vector(u,1,NP);
    free_vector(rhs,1,NP);
    free_vector(a,1,NP);
    free_vector(b,1,NP);
    free_vector(c,1,NP);
    free_matrix(f,1,NP,1,NP);
    return(f[1][array_price]);
}
```

C.4 MacMillan-Zhang method

```
/**************************************************/
/*        MacMillan-Zhang's analytical method     */
/*           for American option prices           */
/*        Written by Byeong-wook Choi             */
/*                October 2000                    */
/**************************************************/
#include <stdio.h>
#include <math.h>
#include <sys/types.h>
#include <unistd.h>
#include "nr.h"
#include "nrutil.h"
#include "jcrit.h"
#define PSMAX   20
#define N      500
#define STEP    10
```

```
#define dt     MATURITY/STEP
#define MATURITY 1.00
#define r      0.08
#define X      50.0
#define sig    0.2236
#define jsig   0.2236
#define lambda 5.0
#define kappa  0.0
/* For Newton-Raphson Method */
#define NN     100
#define NBMAX  50
#define X1     -20.0
#define X2     20
double eta; /* Global variable */
float efunc(float x);
float efunc1(float x);
static float fx(float x)
{
    return efunc(x);
}
static void funcd(float x,float *fn, float *df)
{
    *fn = efunc (x);
    *df = efunc1(x);
}
main(void)
{
    int i,j,k;
    float fixpoint(float);
    float joption(float, float);
    float init_value, out_value;
    float beta1;
    float S_0, value;
    /* For Newton-Raphson Method */
    int nb=NBMAX;
    float xacc,root,*xb1,*xb2;
    /*****************/
    /* STEP I      */
    /*****************/
```

```
    xb1=vector(1,NBMAX);
    xb2=vector(1,NBMAX);
    zbrak(fx,X1,X2,NN,xb1,xb2,&nb);
    for (i=1;i<=nb;i++) {
        xacc=(1.0e-6)*(xb1[i]+xb2[i])/2.0;
        root=rtsafe(funcd,xb1[i],xb2[i],xacc);
        if(root < 0) eta=root;
    }
    free_vector(xb2,1,NBMAX);
    free_vector(xb1,1,NBMAX);
    /*****************/
    /* STEP II      */
    /*****************/
    init_value = X/2.0;
    for (i=1;i<=1000;i++) {
        out_value = fixpoint(init_value);
        if (fabs(init_value-out_value) <1.0e-6) break;
        init_value = out_value;
    }
    printf("After %d Inter., Fixed point (%f) found !!\n",i,out_value);
    /*****************/
    /* STEP III     */
    /*****************/
    beta1 = X-out_value-joption(out_value,MATURITY);
    for (i=0; i<4; i++) {
        S_0 = 40.0 + 5*i;
        value = (S_0 > out_value)?
            joption(S_0,MATURITY)+beta1*pow(S_0/out_value,eta)
              : X - S_0;
        printf("Mac-Zhang option price = %f (euro_price = %f)\n",
            value, joption(S_0, MATURITY));
    }
}
float efunc(float y)
{
    float ans;
    float delta, m;
    delta = sqrt(log(1+pow(jsig,2)));
    m = -0.5*pow(delta,2);
```

```
    ans = pow(sig,2)/2.*pow(y,2) + (r-pow(sig,2)/2.)*y
        - (r+lambda+1./MATURITY)
        + lambda*exp(pow(y,2)*pow(delta,2)/2.+y*m);
    return ans;
}
float efunc1(float y)
{
    double ans;
    float delta, m;
    delta = sqrt(log(1+pow(jsig,2)));
    m = -0.5*pow(delta,2);
    ans = pow(sig,2)*y + (r-pow(sig,2)/2)+lambda*(pow(delta,2)*y+m)
        *exp(pow(y,2)*pow(delta,2)/2+y*m);
    return ans;
}
float fixpoint(float y)
{
    float jeuro_der1(float);
    float joption(float,float);
    extern double eta;
    double ans;
    ans = -eta*(X-joption(y,MATURITY))/(jeuro_der1(y)+1-eta) ;
    return ans;
}
float jeuro_der1(float S)
{
    float nprob(float);
    double poisson_val(float, int);
    double fact(int);
    double dd1;
    double jump_r, jump_sig;
    double sum;
    int i;
    sum=0;
    for(i=0; i<PSMAX; i++) {
        jump_r = r-lambda*kappa + i*log(1+kappa)/MATURITY;
        jump_sig = sqrt(pow(sig,2)+i*pow(jsig,2)/MATURITY);
        dd1 = (log(S)-log(X)+(jump_r+0.5*pow(jump_sig,2))*MATURITY)
            /(jump_sig*sqrt(MATURITY));
```

```
        sum += poisson_val(lambda*MATURITY,i)*(nprob(dd1)-1);
    }
    return(sum);
}
float jeuro_der2(float S)
{
    float nprob(float);
    double poisson_val(float, int);
    double fact(int);
    double dd1;
    double jump_r, jump_sig;
    double sum;
    int i;
    sum=0;
    for(i=0; i<PSMAX; i++) {
        jump_r = r-lambda*kappa + i*log(1+kappa)/MATURITY;
        jump_sig = sqrt(pow(sig,2)+i*pow(jsig,2)/MATURITY);
        dd1 = (log(S)-log(X)+(jump_r+0.5*pow(jump_sig,2))*MATURITY)
            /(jump_sig*sqrt(MATURITY));
        sum += poisson_val(lambda*MATURITY,i)
            *(1/sqrt(2*3.14159)*exp(-pow(dd1,2)/2))
            *(1/S)*(1/jump_sig)/sqrt(MATURITY);
    }
    return(sum);
}
```

C.5 Modified MacMillan-Zhang method

```
/************************************************/
/*       Modified MacMillan-Zhang's method      */
/*          for American option prices          */
/*       Written by Byeong-wook Choi            */
/*               October 2000                   */
/************************************************/
#include <stdio.h>
#include <math.h>
#include <sys/types.h>
#include <unistd.h>
```

```
#include "nr.h"
#include "nrutil.h"
#define PSMAX    20
#define N       500
#define STEP     10
#define dt    MATURITY/STEP
#define MATURITY 1.00
#define r        0.06
#define X        100.0
#define sig      0.3
#define jsig     0.15
#define kappa    0.0
#define lambda   1.0
/* For Newton-Raphson Method */
#define NN       100
#define NBMAX    50
#define X1       -20.0
#define X2       20
/* Global variable */
double eta;
double maturity;
float efunc(float x);
float efunc1(float x);
static float fx(float x)
{
    return efunc(x);
}
static void funcd(float x,float *fn, float *df)
{
    *fn = efunc (x);
    *df = efunc1(x);
}
main(void)
{
    int i,j,k;
    float fixpoint(float);
    float joption(float, float);
    float jprob(float, float,float);
    float crit_price[N];
```

```
float price_1, sum_price_n;
float init_value, out_value;
float beta1;
float S_0, value;
/* For Newton-Raphson Method */
int nb=NBMAX;
float xacc,root,*xb1,*xb2;
for(j=1; j<=STEP; j++) {
    maturity = MATURITY*j/STEP;
    /*****************/
    /* STEP I        */
    /*****************/
    xb1=vector(1,NBMAX);
    xb2=vector(1,NBMAX);
    zbrak(fx,X1,X2,NN,xb1,xb2,&nb);
    for (i=1;i<=nb;i++) {
        /* We can 1.0e-6 -> 1.0e-5, if we want STEP >= 20 */
        xacc=(1.0e-6)*(xb1[i]+xb2[i])/2.0;
        root=rtsafe(funcd,xb1[i],xb2[i],xacc);
        if(root < 0) eta=root;
    }
    free_vector(xb2,1,NBMAX);
    free_vector(xb1,1,NBMAX);
    /*****************/
    /* STEP II       */
    /*****************/
    init_value = X/2.0;
    for (i=1;i<=5000;i++) {
        out_value = fixpoint(init_value);
        if (fabs(init_value-out_value) <1.0e-6) break;
        init_value = out_value;
    }
    crit_price[j] = out_value;
}
/*****************/
/* STEP III      */
/*****************/
/* Computing price_n */
for(i=0; i<5; i++) {
```

```
        S_0 = 80.0 +i*10;
        price_1 = joption(S_0, MATURITY);
        sum_price_n = price_1
             + 0.5*dt*r*X*exp(-r*MATURITY)*jprob(S_0, X, MATURITY);
        for(k=1; k<STEP; k++) {
            sum_price_n += (r*X*MATURITY/STEP) *
                       ( exp(-r*(STEP-k)*dt)
                     *jprob(S_0, crit_price[k], (STEP-k)*dt));
        }
    }
}
float fixpoint(float y)
{
    float jeuro_der1(float);
    float joption(float,float);
    extern double eta;
    extern double maturity;
    double ans;
    ans = -eta*(X-joption(y,maturity))/(jeuro_der1(y)+1-eta) ;
    return ans;
}
float efunc(float y)
{
    extern double maturity;
    float ans;
    float delta, m;
    delta = sqrt(log(1+pow(jsig,2)));
    m = -0.5*pow(delta,2);
    ans = pow(sig,2)/2.*pow(y,2) + (r-pow(sig,2)/2.)*y
        -(r+lambda+1./maturity)
+ lambda*exp(pow(y,2)*pow(delta,2)/2.+y*m);
    return ans;
}
float efunc1(float y)
{
    double ans;
    float delta, m;
    delta = sqrt(log(1+pow(jsig,2)));
    m = -0.5*pow(delta,2);
```

```
    ans = pow(sig,2)*y + (r-pow(sig,2)/2)+lambda*(pow(delta,2)*y+m)
        *exp(pow(y,2)*pow(delta,2)/2+y*m);
    return ans;
}
float jeuro_der1(float S)
{
    extern double maturity;
    float nprob(float);
    double poisson_val(float, int);
    double fact(int);
    double dd1;
    double jump_r, jump_sig;
    double sum;
    int i;
    sum=0;
    for(i=0; i<PSMAX; i++) {
        jump_r = r-lambda*kappa + i*log(1+kappa)/maturity;
        jump_sig = sqrt(pow(sig,2)+i*pow(jsig,2)/maturity);
        dd1 = (log(S)-log(X)+(jump_r+0.5*pow(jump_sig,2))*maturity)
            /(jump_sig*sqrt(maturity));

        sum += poisson_val(lambda*maturity,i)*(nprob(dd1)-1);
    }
    return(sum);
}
float jeuro_der2(float S)
{
    float nprob(float);
    double poisson_val(float, int);
    double fact(int);
    double dd1;
    double jump_r, jump_sig;
    double sum;
    int i;
    sum=0;
    for(i=0; i<PSMAX; i++) {
        jump_r = r-lambda*kappa + i*log(1+kappa)/MATURITY;
        jump_sig = sqrt(pow(sig,2)+i*pow(jsig,2)/MATURITY);
        dd1 = (log(S)-log(X)+(jump_r+0.5*pow(jump_sig,2))*MATURITY)
```

```
            /(jump_sig*sqrt(MATURITY));
        sum += poisson_val(lambda*MATURITY,i)
            *(1/sqrt(2*3.14159)*exp(-pow(dd1,2)/2))
            *(1/S)*(1/jump_sig)/sqrt(MATURITY);
    }
    return(sum);
}
```

C.6 Extended Ju's method

```
/********************************************************/
/*    Modified JU's Method for American put prices      */
/*       under jump-diffusion processes                 */
/*           Written by Byeong-wook Choi                */
/*                 June  2001                           */
/********************************************************/
#include <stdio.h>
#include <math.h>
#include <time.h>
#include <sys/types.h>
#include <unistd.h>
#include "nr.h"
#include "nrutil.h"
#include "jcrit.h"
#define STEP  100
#define dt    MAT/STEP
#define DIM   2
#define NN    100
#define NBMAX 50
#define X1    -20.0
#define X2    20
#define PSMAX 20
/* An example */
#define MAT   0.25
#define r     0.06
#define X     100.0
#define sig   0.3
#define lambda   1.0
#define kappa    0.0
```

```
#define jsig     0.15
#define div      0
/* Global variables */
double eta;
float B11, B21, B22, B31, B32, B33;
float b11, b21, b22, b31, b32, b33;

float joption(float,float);
float jprob(float,float,float);
float joption_d(float,float);
float jprob_d(float,float,float);
float fixpoint(float);
float efunc(float x);
float efunc1(float x);

static float fx(float x)
{
    return efunc(x);
}
static void funcd(float x,float *fn, float *df)
{
    *fn = efunc (x);
    *df = efunc1(x);
}
/* Define the function solved */
void funcv1(int n,float x[],float f[])
{
    float eep1(float, float);
    float eep1_d(float, float);
    f[1]=joption(x[1],MAT)+eep1(x[1],x[2])-X+x[1];
    f[2]=joption_d(x[1],MAT)+eep1_d(x[1],x[2])+1;
}
float eep1(float x1, float x2)
{
    int k;
    float sum;
    sum=0;
    for(k=1; k<STEP; k++) {
        sum += r*dt*X*exp(-r*k*dt)*jprob(x1,x1*exp(x2*k*dt),k*dt);
```

```
    }
    sum += 0.5*r*dt*X*exp(-r*MAT)*jprob(x1, X, MAT);
    return(sum);
}
float eep1_d(float x1, float x2)
{
    int k;
    float sum;
    sum=0;
    for(k=1; k<STEP; k++) {
        sum += r*dt*X*exp(-r*k*dt)*jprob_d(x1,x1*exp(x2*k*dt),k*dt);
    }
    sum += 0.5*r*dt*X*exp(-r*MAT)*jprob_d(x1,X,MAT);
    return(sum);
}
void funcv21(int n,float x[],float f[])
{
    float eep21(float, float);
    float eep21_d(float, float);
    f[1]=joption(x[1]*exp(x[2]*MAT/2),MAT/2.) + eep21(x[1],x[2])
        -X+x[1]*exp(x[2]*MAT/2);
    f[2]=joption_d(x[1]*exp(x[2]*MAT/2),MAT/2.) +eep21_d(x[1],x[2])+1;
}
float eep21(float x1, float x2)
{
    int k;
    float sum;
    sum=0;
    for(k=STEP/2+1; k<STEP; k++) {
        sum += r*dt*X*exp(-r*(k-STEP/2)*dt)
            *jprob(x1*exp(x2*MAT/2),x1*exp(x2*k*dt),(k-STEP/2)*dt);
    }
    sum += 0.5*r*dt*X*exp(-r*MAT/2.)*jprob(x1*exp(x2*MAT/2), X, MAT/2.);
    return(sum);
}
float eep21_d(float x1, float x2)
{
    int k;
    float sum;
```

```
    sum=0;
    for(k=STEP/2+1; k<STEP; k++) {
        sum += r*dt*X*exp(-r*(k-STEP/2)*dt)
            *jprob_d(x1*exp(x2*MAT/2), x1*exp(x2*k*dt),(k-STEP/2)*dt);
    }
    sum += 0.5*r*dt*X*exp(-r*MAT/2.)*jprob_d(x1*exp(x2*MAT/2),X,MAT/2.);
    return(sum);
}
void funcv22(int n, float x[], float f[])
{
    float eep22(float ,float);
    float eep22_d(float, float);
    f[1] = joption(x[1],MAT) + eep22(x[1],x[2]) -X+x[1];
    f[2] = joption_d(x[1],MAT) + eep22_d(x[1],x[2]) +1;
}
float eep22(float x1, float x2)
{
    int k;
    float sum;
    sum=0;
    for(k=1; k<=STEP/2; k++) {
        sum += r*dt*X*exp(-r*k*dt)*jprob(x1,x1*exp(x2*k*dt),k*dt);
    }
    for(k=STEP/2+1; k<STEP; k++) {
        sum += r*dt*X*exp(-r*k*dt)*jprob(x1,B21*exp(b21*k*dt),k*dt);
    }
    sum += 0.5*r*dt*X*exp(-r*MAT)*jprob(x1, X, MAT);
    return(sum);
}
float eep22_d(float x1, float x2)
{
    int k;
    float sum;
    sum=0;
    for(k=1; k<=STEP/2; k++) {
        sum += r*dt*X*exp(-r*k*dt)*jprob_d(x1,x1*exp(x2*k*dt),k*dt);
    }
    for(k=STEP/2+1; k<STEP; k++) {
        sum += r*dt*X*exp(-r*k*dt)*jprob_d(x1,B21*exp(b21*k*dt),k*dt);
```

```
    }
    sum += 0.5*r*dt*X*exp(-r*MAT)*jprob_d(x1, X, MAT);
    return(sum);
}
void funcv31(int n, float x[], float f[])
{
    float eep31(float, float);
    float eep31_d(float, float);
    f[1]=joption(x[1]*exp(x[2]*2*MAT/3),MAT/3.) + eep31(x[1],x[2])
        -X+x[1]*exp(x[2]*2*MAT/3);
    f[2]=joption_d(x[1]*exp(x[2]*2*MAT/3),MAT/3.) + eep31_d(x[1],x[2])+1;
}
float eep31(float x1, float x2)
{
    int k;
    float BB31;
    float sum;
    BB31=x1*exp(x2*2*MAT/3);
    sum=0;
    for(k=1; k<STEP/3; k++) {
        sum += r*dt*X*exp(-r*k*dt)*jprob(BB31,BB31*exp(x2*k*dt),k*dt);
    }
    sum += 0.5*r*dt*X*exp(-r*MAT/3.)*jprob(BB31, X, MAT/3.);
    return(sum);
}
float eep31_d(float x1, float x2)
{
    int k;
    float BB31;
    float sum;
    BB31=x1*exp(x2*2*MAT/3);
    sum=0;
    for(k=1; k<STEP/3; k++) {
        sum += r*dt*X*exp(-r*k*dt)*jprob_d(BB31,BB31*exp(x2*k*dt),k*dt);
    }
    sum += 0.5*r*dt*X*exp(-r*MAT/3.)*jprob_d(BB31, X, MAT/3.);
    return(sum);
}
void funcv32(int n, float x[], float f[])
```

```
{
    float eep32(float, float);
    float eep32_d(float, float);
    f[1]=joption(x[1]*exp(x[2]*MAT/3),2*MAT/3.) + eep32(x[1],x[2])
        -X+x[1]*exp(x[2]*MAT/3);
    f[2]=joption_d(x[1]*exp(x[2]*MAT/3),2*MAT/3.) + eep32_d(x[1],x[2])+1;
}
float eep32(float x1, float x2)
{
    int k;
    float BB32;
    float sum;
    BB32=x1*exp(x2*MAT/3);
    sum=0;
    for(k=STEP/3+1; k<=2*STEP/3; k++) {
        sum += r*dt*X*exp(-r*(k-STEP/3)*dt)
            *jprob(BB32, x1*exp(x2*k*dt),(k-STEP/3)*dt);
    }
    for(k=2*STEP/3+1; k<STEP; k++) {
        sum += r*dt*X*exp(-r*(k-STEP/3)*dt)
            *jprob(BB32, B31*exp(b31*k*dt),(k-STEP/3)*dt);
    }
    sum += 0.5*r*dt*X*exp(-r*2*MAT/3.)*jprob(BB32, X, 2*MAT/3.);
    return(sum);
}
float eep32_d(float x1, float x2)
{
    int k;
    float BB32;
    float sum;
    BB32=x1*exp(x2*MAT/3);
    sum=0;
    for(k=STEP/3+1; k<=2*STEP/3; k++) {
        sum += r*dt*X*exp(-r*(k-STEP/3)*dt)
            *jprob_d(BB32, x1*exp(x2*k*dt),(k-STEP/3)*dt);
    }
    for(k=2*STEP/3+1; k<STEP; k++) {
        sum += r*dt*X*exp(-r*(k-STEP/3)*dt)
            *jprob_d(BB32, B31*exp(b31*k*dt),(k-STEP/3)*dt);
```

```
    }
    sum += 0.5*r*dt*X*exp(-r*2*MAT/3.)*jprob_d(BB32, X, 2*MAT/3.);
    return(sum);
}
void funcv33(int n, float x[], float f[])
{
    float eep33(float ,float);
    float eep33_d(float, float);
    f[1] = joption(x[1],MAT) + eep33(x[1],x[2]) -X+x[1];
    f[2] = joption_d(x[1],MAT) + eep33_d(x[1],x[2]) +1;
}
float eep33(float x1, float x2)
{
    int k;
    float sum;
    sum=0;
    for(k=1; k<=STEP/3; k++) {
        sum += r*dt*X*exp(-r*k*dt)*jprob(x1,x1*exp(x2*k*dt),k*dt);
    }
    for(k=STEP/3+1; k<=2*STEP/3; k++) {
        sum += r*dt*X*exp(-r*k*dt)*jprob(x1,B32*exp(b32*k*dt),k*dt);
    }
    for(k=2*STEP/3+1; k<STEP; k++) {
        sum += r*dt*X*exp(-r*k*dt)*jprob(x1,B31*exp(b31*k*dt),k*dt);
    }
    sum += 0.5*r*dt*X*exp(-r*MAT)*jprob(x1, X, MAT);
    return(sum);
}
float eep33_d(float x1, float x2)
{
    int k;
    float sum;
    sum=0;
    for(k=1; k<=STEP/3; k++) {
        sum += r*dt*X*exp(-r*k*dt)*jprob_d(x1,x1*exp(x2*k*dt),k*dt);
    }
    for(k=STEP/3+1; k<=2*STEP/3; k++) {
        sum += r*dt*X*exp(-r*k*dt)*jprob_d(x1,B32*exp(b32*k*dt),k*dt);
    }
```

```
    for(k=2*STEP/3+1; k<STEP; k++) {
        sum += r*dt*X*exp(-r*k*dt)*jprob_d(x1,B31*exp(b31*k*dt),k*dt);
    }
    sum += 0.5*r*dt*X*exp(-r*MAT)*jprob_d(x1, X, MAT);
    return(sum);
}

/********************************************/
/*                                          */
/*   M A I N   P R O G R A M                */
/*                                          */
/********************************************/
int main(void)
{
    int i,k,check;
    float *x,*f;
    /* For Newton-Raphson Method */
    int nb=NBMAX;
    float xacc,root,*xb1,*xb2;
    float init_value, out_value;

    float S0;        /* initial stock price */
    float P1, P2, P3;    /* prices of option */
    float Euro_price;
    float step, ddt;    /* ddt=MATURITY/step. */
    /*****************/
    /* Prem: Step1   */
    /*****************/
    xb1=vector(1,NBMAX);
    xb2=vector(1,NBMAX);
    zbrak(fx,X1,X2,NN,xb1,xb2,&nb);
    for (i=1;i<=nb;i++) {
        xacc=(1.0e-6)*(xb1[i]+xb2[i])/2.0;
        root=rtsafe(funcd,xb1[i],xb2[i],xacc);
        if(root < 0) eta=root;
    }
    free_vector(xb2,1,NBMAX);
    free_vector(xb1,1,NBMAX);
    /*****************/
```

```
/* Prem: STEP II */
/*****************/
init_value = X/2.0;
for (i=1;i<=1000;i++) {
    out_value = fixpoint(init_value);
    if (fabs(init_value-out_value) <1.0e-6) break;
    init_value = out_value;
}
printf("** Preliminary **\n");
printf("Searching initial solution\n");
printf("using Macmillan's method:\n");
printf("After %d Interation, Fixed point (%f) found !!\n",i,out_value);
/***************************/
/*      P A R T   I        */
/***************************/
/* Initialize */
x=vector(1,DIM);
f=vector(1,DIM);
/* Initial guessing of solution */
/* -------------------------------------*/
/*    NOTE:                             */
/* Different initial guesses might yield */
/* an error message: convergence problem */
/* -------------------------------------*/
x[1]=out_value;
x[2]=0.0;
broydn(x,DIM,&check,funcv1);
funcv1(DIM,x,f);
if (check) printf("Convergence problems.\n");
B11 = x[1];
b11 = x[2];
printf("%7s %3s %12s\n","Index","x","f");
for (i=1;i<=DIM;i++) printf("%5d %12.6f %12.6f\n",i,x[i],f[i]);
/***************************/
/*      P A R T   II - 1   */
/***************************/
broydn(x,DIM,&check,funcv21);
funcv21(DIM,x,f);
if (check) printf("Convergence problems.\n");
```

```
B21 = x[1];
b21 = x[2];
printf("%7s %3s %12s\n","Index","x","f");
for (i=1;i<=DIM;i++) printf("%5d %12.6f %12.6f\n",i,x[i],f[i]);
/****************************/
/*      P A R T   II - 2   */
/****************************/
broydn(x,DIM,&check,funcv22);
funcv22(DIM,x,f);
if (check) printf("Convergence problems.\n");
B22 = x[1];
b22 = x[2];
printf("%7s %3s %12s\n","Index","x","f");
for (i=1;i<=DIM;i++) printf("%5d %12.6f %12.6f\n",i,x[i],f[i]);
/****************************/
/*      P A R T   III - 1  */
/****************************/
broydn(x,DIM,&check,funcv31);
funcv31(DIM,x,f);
if (check) printf("Convergence problems.\n");
B31 = x[1];
b31 = x[2];
printf("%7s %3s %12s\n","Index","x","f");
for (i=1;i<=DIM;i++) printf("%5d %12.6f %12.6f\n",i,x[i],f[i]);
/****************************/
/*      P A R T   III - 2  */
/****************************/
broydn(x,DIM,&check,funcv32);
funcv32(DIM,x,f);
if (check) printf("Convergence problems.\n");
B32 = x[1];
b32 = x[2];
printf("%7s %3s %12s\n","Index","x","f");
for (i=1;i<=DIM;i++) printf("%5d %12.6f %12.6f\n",i,x[i],f[i]);
/****************************/
/*      P A R T   III - 3  */
/****************************/
broydn(x,DIM,&check,funcv33);
funcv33(DIM,x,f);
```

```
if (check) printf("Convergence problems.\n");
B33 = x[1];
b33 = x[2];
printf("%7s %3s %12s\n","Index","x","f");
for (i=1;i<=DIM;i++) printf("%5d %12.6f %12.6f\n",i,x[i],f[i]);
free_vector(f,1,DIM);
free_vector(x,1,DIM);
/****************************/
/* Computing Price(P1,P2,P3) */
/****************************/
S0 = 120;    /* NEED TO CHANGE 5/19/2002 */
Euro_price = joption(S0,MAT);
step = 100.0;
ddt = MAT/step;
/** PRICE P1  **/
P1 = Euro_price + 0.5*r*X*ddt * exp(-r*MAT) * jprob(S0, X, MAT);
for(k=1; k<step; k++) {
    P1 += (r*X*ddt) * exp(-r*k*ddt)
            * jprob(S0, B11*exp(b11*k*ddt), k*ddt);
}
/** PRICE P2  **/
P2 = Euro_price + 0.5*r*X*ddt * exp(-r*MAT) * jprob(S0, X, MAT);
for(k=1; k<step/2; k++) {
    P2+= (r*X*ddt) * exp(-r*k*ddt)
                * jprob(S0, B22*exp(b22*k*ddt), k*ddt);
}
for(k=step/2; k<step; k++) {
    P2 += (r*X*ddt) * exp(-r*k*ddt)
            * jprob(S0, B21*exp(b21*k*ddt), k*ddt);
}
/** PRICE P3  **/
P3 = Euro_price + 0.5*r*X*ddt * exp(-r*MAT) * jprob(S0, X, MAT);
for(k=1; k<step/3; k++) {
    P3+= (r*X*ddt) * exp(-r*k*ddt)
               * jprob(S0, B33*exp(b33*k*ddt), k*ddt);
}
for(k=step/3; k<2*step/3; k++) {
    P3 += (r*X*ddt) * exp(-r*k*ddt)
            * jprob(S0, B32*exp(b32*k*ddt), k*ddt);
```

```
    }
    for(k=2*step/3; k<step; k++) {
        P3 += (r*X*ddt) * exp(-r*k*ddt)
                * jprob(S0, B31*exp(b31*k*ddt), k*ddt);
    }
    printf("Euro_price = %f\n", Euro_price);
    printf("P1=%f\n", P1);
    printf("P2=%f\n", P2);
    printf("P3=%f\n", P3);
    printf("Extrapolated price=%f\n", 4.5*P3-4*P2+0.5*P1);
    return 0;
}

float efunc(float y)
{
    float ans;
    float delta, m;
    delta = sqrt(log(1+pow(jsig,2)));
    m = -0.5*pow(delta,2);
    ans = pow(sig,2)/2.*pow(y,2)
        + (r-pow(sig,2)/2.)*y-(r+lambda+1./MAT)
        + lambda*exp(pow(y,2)*pow(delta,2)/2.+y*m);
    return ans;
}
float efunc1(float y)
{
    double ans;
    float delta, m;
    delta = sqrt(log(1+pow(jsig,2)));
    m = -0.5*pow(delta,2);
    ans = pow(sig,2)*y+(r-pow(sig,2)/2)+lambda*(pow(delta,2)*y+m)
        *exp(pow(y,2)*pow(delta,2)/2+y*m);
    return ans;
}
float jeuro_der1(float S)
{
    float nprob(float);
    double poisson_val(float, int);
    double fact(int);
```

```
    double dd1;
    double jump_r, jump_sig;
    double sum;
    int i;
    sum=0;
    for(i=0; i<PSMAX; i++) {
        jump_r = r-lambda*kappa + i*log(1+kappa)/MAT;
        jump_sig = sqrt(pow(sig,2)+i*pow(jsig,2)/MAT);
        dd1 = (log(S)-log(X)+(jump_r+0.5*pow(jump_sig,2))*MAT)
            /(jump_sig*sqrt(MAT));
        sum += poisson_val(lambda*MAT,i)*(nprob(dd1)-1);
    }
    return(sum);
}
float jeuro_der2(float S)
{
    float nprob(float);
    double poisson_val(float, int);
    double fact(int);
    double dd1;
    double jump_r, jump_sig;
    double sum;
    int i;
    sum=0;
    for(i=0; i<PSMAX; i++) {
        jump_r = r-lambda*kappa + i*log(1+kappa)/MAT;
        jump_sig = sqrt(pow(sig,2)+i*pow(jsig,2)/MAT);
        dd1 = (log(S)-log(X)+(jump_r+0.5*pow(jump_sig,2))*MAT)
            /(jump_sig*sqrt(MAT));
        sum += poisson_val(lambda*MAT,i)
            *(1/sqrt(2*3.14159)*exp(-pow(dd1,2)/2))
            *(1/S)*(1/jump_sig)/sqrt(MAT);
    }
    return(sum);
}
float fixpoint(float y)
{
    float jeuro_der1(float);
    float joption(float,float);
```

```
    extern double eta;
    double ans;
    ans = -eta*(X-joption(y,MAT))/(jeuro_der1(y)+1-eta) ;
    return ans;
}
/**********************************************/
/*      - Calculation of Integration          */
/**********************************************/
#define    nof(x)    1/sqrt(2*4.0e0*atan(1.e0))*exp(-pow(x,2)/2)
float Integ(float t1, float t2, float S, float B,
        float b,float num1,float num2)
{
    double nprob(float);
    float z1,z2,z3;
    float value;
    z1 = (r-div-b+num1*SQR(sig)/2)/sig;
    z2 = (log(S)-log(B))/sig;
    z3 = sqrt(SQR(z1)+2*num2);
    value = exp(-num2*t1)*nprob(z1*sqrt(t1)+z2/sqrt(t1))
        -exp(-num2*t2)*nprob(z1*sqrt(t2)+z2/sqrt(t2))
        +0.5*(z1/z3+1)*exp(z2*(z3-z1))
            *(nprob(z3*sqrt(t2)+z2/sqrt(t2))
              -nprob(z3*sqrt(t1)+z2/sqrt(t1)))
        +0.5*(z1/z3-1)*exp(-z2*(z3+z1))
            *(nprob(z3*sqrt(t2)-z2/sqrt(t2))
               -nprob(z3*sqrt(t1)-z2/sqrt(t1)));
    return(value);
}
float Integ_der(float t1, float t2, float S, float B,
        float b, float num1, float num2)
{
    double nprob(float);
    float z1,z2,z3;
    float valued;
    z1 = (r-div-b+num1*SQR(sig)/2)/sig;
    z2 = (log(S)-log(B))/sig;
    z3 = sqrt(SQR(z1)+2*num2);
    valued = 1/(sig*S)*(exp(-num2*t1)/sqrt(t1)
            *nof(z1*sqrt(t1)+z2/sqrt(t1))
```

```
            -exp(-num2*t2)/sqrt(t2)*nof(z1*sqrt(t2)+z2/sqrt(t2)))
        +0.5*(z1/z3+1)/(sig*S)*exp(z2*(z3-z1))
            *(nprob(z3*sqrt(t2)+z2/sqrt(t2))
            -nprob(z3*sqrt(t1)+z2/sqrt(t1)))*(z3-z1)
        +0.5*(z1/z3+1)/(sig*S)*exp(z2*(z3-z1))
            *(nof(z3*sqrt(t2)+z2/sqrt(t2))/sqrt(t2)
            -nof(z3*sqrt(t1)+z2/sqrt(t1))/sqrt(t1))
        -0.5*(z1/z3-1)/(sig*S)*exp(-z2*(z3+z1))
            *(nprob(z3*sqrt(t2)-z2/sqrt(t2))
            -nprob(z3*sqrt(t1)-z2/sqrt(t1)))*(z3+z1)
        -0.5*(z1/z3-1)/(sig*S)*exp(-z2*(z3+z1))
            *(nof(z3*sqrt(t2)-z2/sqrt(t2))/sqrt(t2)
            -nof(z3*sqrt(t1)-z2/sqrt(t1))/sqrt(t1));
    return(valued);
}
```

Bibliography

[1] Ahn, C. (1992): Opion Pricing When Jump Risk is Systematic, *Mathematical Finance*, 2, pp. 299-308.

[2] Ahn, C. and H. Thomson (1988): Jump-Diffusion Processes and the Term Structure of Interest Rates, *Journal of Finance*, 43, pp. 155-174.

[3] Amin, K. (1933): Jump-Diffusion Option Valuation in Discrete-Time, *Journal of Finance*, 48, pp. 1833-1863.

[4] Anderson, L., and M. Broadie (2001): A Primal-Dual Simulation Algorithm for Pricing Multi-Dimensional American Options, working paper, Columbia University.

[5] Attari, M. (1999): Discontinuous Interest Rate Processes: An Equilibrium Model for Bond Option Prices, *Journal of Financial and Quantitative Analysis*, 34, pp. 293-322.

[6] Ball, C., and W. Torous (1985): On Jumps in Common Stock Prices and Their Impact on Call Option Pricing, *The Journal of Finance*, 40, pp. 155 - 173.

[7] Barone-Adesi, G. and R. Whaley (1987): Efficienct Analytical Approximation of American Option Values, *Journal of Finance*, 42, pp. 301-320.

[8] Bates, D. (1996): Jumps and Stochastic Volatility: Exchange Rate Processes Implicit in Deutche Mark Options, *The Review of Financial Studies*, 9, pp. 69-107.

[9] Bertsimas D., L. Kogan, and Lo, A.W. (2001): Pricing and Hedging Derivative Securities in Incomplete Markets, *Operations Research*, Vol. 49, No. 3. pp. 372-397.

[10] Black, F. and M. Scholes (1973): The Pricing of Options and Corporate Liabilities, *Journal of Political Economy*, 81, pp. 637-659.

[11] Boyle, P., Options: A Monte Carlo Approach (1977): *Journal of Financial Economics*, 4, pp. 323-338.

[12] Boyle, P. (1988): A Lattice Framework for Option Pricing with Two State Variables, *Journal of Financial and Quantitative Analysis*, 23, pp. 1-12.

[13] Broadie, M. and J. Detemple (1997): The Valuation of American Options on Multiple Assets, *Mathematical Finance*, 7, pp. 241-286.

[14] Boyle, P.P., J. Evnine, and S. Gibbs (1989): Numerical Evaluation of Multivariate Contingent Claims, *Review of Financial Studies*, 2, pp. 241-250.

[15] Broadie, M. and P. Glasserman (1996): Estimating Security Price Derivatives using Simulation, *Management Science*, 42, pp. 269-285.

[16] Broadie, M. and P. Glasserman (1997a): Pricing American Style Securities Using Simulation, *Journal of Economic Dynamics and Control*, 21, 8-9, pp. 1323-1352.

[17] Broadie, M. and P. Glasserman (1997b): A Stochastic Mesh Method for Pricing High-Dimensional American Options, working paper, Columbia University.

[18] Brealey, R. A., and S. C. Myers (1991): *Principles of Corporate Finance*, Fourth Edition, McGraw-Hill, Inc.

[19] Brennan M., and E. Schwartz (1977): The Valuation of American Put Options, *Journal of Finance*, 32, pp. 449-462.

[20] Broadie, M., and Jerome Detemple (1996): American Option Valuation: New Bounds, Approximations, and a Comparison of Existing Methods, *The Review of Financial Studies*, 9, pp. 1211-1250.

[21] Campbell, J., A. Lo, and A. C. Mackinlay (1997): *The Econometrics of Financial Markets*, Princeton University Press.

[22] Carr, P., R. Jarrow, and R. Myneni (1992): Alternative Characterizations of American Puts, *Mathematical Finance*, 2, pp. 87-106.

[23] Cochrane, J. (2001) *Asset Pricing*, 2001, Princeton University Press.

[24] Colwell D.B., R.J. Elliott (1993): Discontinuous Asset Prices and Non-Attainable Contingent Claims, *Mathematical Finance*, 3, pp. 295-308.

[25] Constantinides, G., and T. Zariphopoulou (1999): Bounds on Prices of Contingent Claims in an Intertemporal Economy with Proportional Transaction Costs and general Preferences, *Finance and Stochastics*, 3, pp. 345-369.

[26] Cox, J. C., S. A. Ross and M. Rubinstein (1979): Option Pricing: A Simplified Approach, *Journal of Financial Economics*, 7, pp. 229-264.

[27] Cryer, C. W. (1983): The Efficient Solution of Linear Complementarity Problems for Tridiagonal Minkowski Matrices, *ACM Trans. Math. Software*, 9, pp. 199-214.

[28] Davis, M.H.A., V.G. Panas, and T. Zariphopoulou (1993): European Option Pricing with Transaction Costs, *SIAM Journal of Control and Optimization*, 31, pp. 470-493.

[29] Davis, M.H.A., and T. Zariphopoulou (1995): American Options and Transaction Fees, *Mathematical Finance*, The IMA Volumes in Mathematics and its applications, V.65, pp. 47-61.

[30] Dempster, M.A.H., and J.P.Hutton (1997): Fast Numerical Valuation of American, Exotics, and Complex Options, *Applied Mathematical Finance*, 4, pp. 1-20

[31] Dempster, M. A., and J. P. Hutton (1999): Pricing American Stock Options by Linear Programming, *Mathematical Finance*, 9, pp. 229-254.

[32] Duffie, D., and M. Jackson (1990): Optimal Hedging and Equilibrium in a Dynamic Futures Market, *Journal of Economic Dynamics and Control*, 14, pp. 21-33.

[33] Duffie, D., and H. Richardson (1991): Mean-Variance Hedging in Continuous Time, *Annals of Applied Probability*, 1, pp. 1-15.

[34] El Karoui, N., and M. Quenez (1995): Dynamic programming and Pricing of Contingent Claims in an Incomplete Market, *SIAM Journal of Control and Optimization*, 33, pp. 29-66.

[35] Fama, E. F. (1970): Efficient Capital Markets: A Review of Theory and Empirical Work, *Journal of Finance*, 25, pp. 383-417.

[36] Fleming, W., and R. Rishel (1975): *Deterministic and Stochastic Optimal Control*, Springer-Verlag.

[37] Föllmer, H., and M. Schweizer (1990): Hedging of Contingent Claims under Incomplete Information, *Applied Stochastic Analysis*, Gordon and Breach, London. pp. 389-414.

[38] Föllmer, H., and M. Sondermann (1986): Hedging of Nonredundant Contingent Claims, *Contributions to Mathematical Economics in Honour of Gerard Debreu*, North Holland. Amsterdam, pp. 205-223.

[39] Geske, R., and H. E. Johnson (1984): The American Put Option Valued Analytically, *The Journal of Finance*, 39, pp. 1511-1524.

[40] Geske, R., and K. Shastri (1985): Valuation by Approximation: a Comparison of Alternative Option Valuation Techniques, *Journal of Financial and Quantitative Analysis*, 20, pp. 45-71.

[41] Gukhal, C. (2001): Analytical Valuation of American Options on Jump-Diffusion Processes, *Mathematical Finance*, 11, pp. 97-115.

[42] Harrison, J. M. and D. M. Kreps (1979): *Martingales and Arbitrage in Multiperiod securities Markets*, Journal of Economic Theory, 20, pp. 381-408.

[43] Harrison, J. M. and S. R. Pliska (1981): *Martingales and Stochastic Integrals in the Theory of Continuous Trading*, Stochasti Processes and their Appliactions, 11, pp. 215-260.

[44] Harrison, J. M. and S. R. Pliska (1983): *A Stochastic Calculus Model of Continuous Trading: Complete Markets*, Stochasti Processes and their Appliactions, 15, pp. 313-316.

[45] Haugh, M., and L. Kogan (2001): Pricing American Options: A Duality Approach, working paper, MIT.

[46] Huang, J., M. Subrahmanyam and G. Yu (1996): Pricing and Hedging American Options: A Recursive Integration Method, *The Review of Financial Studies*, 9, pp. 277-300.

[47] Ingersoll, J.E., Jr. (1987): *Theory of Financial Decision Making*, Rowman & Littlefield, Totowa, N.J.

[48] Jacka, S. (1991): Optimal Stopping and the American Put, *Mathematical Finance*, 1, pp.1-14.

[49] Jaillet, P., D. Lamberton and B. Lapeyre (1990): Variational Inequalities and the Pricing of American Options, *Acta Applicandae Mathematicae*, 21, pp. 263-289.

[50] Jarrow, R., and E. Rosenfeld (1984): Jump Risks and the Intertemporal Capital Asset Pricing Model, *Journal of Business*, 57, pp. 337 - 351.

[51] Jarrow, R., and A. Rudd (1983): *Option Pricing*, Dow Jones-Irwin.

[52] Jeanblanc-Picque, M., and M. Pontier (1990): Optimal Portfolio for a Small Investor in a Market Model with Discontinuous Prices, *Applied Mathematics and Optimization*, 22, pp. 287-310.

[53] Johnson, H (1983): An Analytical Approximation for the American Put Price, *Journal of Financial and Quantitative Analysis*, 18, pp. 141-148.

[54] Jorion, P. (1988): On Jumps Processes in the Foreign Exchange and Stock Markets, *The Review of Financial Studies*, 1, pp. 427 - 445.

[55] Ju, N. (1998): Pricing an American Option by Approximating Its Early Exercise Boundary as a Multipiece Exponential Function, *The Review of Financial Studies*, 11, pp. 647-674.

[56] Ju, N. and R. Zhong (1999): An Approximate Formula for Pricing American Options, *The Journal of Derivatives*, pp. 31-40.

[57] Kim, I. J (1990): The Analytical Valuation of American Options, *The Review of Financial Studies*, 3, pp. 547-572.

[58] Kremer, J. and R. Rosenfeldt (1992): Warrant Pricing: Jump-Diffusion vs. Black-Scholes, *Journal of Financial and Quantitative Analysis*, 28, pp. 255-272.

[59] Kushner, A. (1995): *Numerical Methods for Stochastic Control Problems in Continuous Time*, Springer-Verlag, Berlin.

[60] Lamberton, D., B. Lapeyre (1997): *Introduction to Stochastic Calculus Applied to Finance*, Chapman & Hall.

[61] Leisen, D. (1998): Pricing the American Put Option: A Detailed Convergence Analysis for Binomial Models, *Journal of Economic Dynamics & Control*, 22, pp. 1419-1444.

[62] Longstaff, F., and E. Schwartz (2001): Valuing American Options by Simulation: a Simplified Least-Square Approach, *The Review of Financial Studies*, 14, pp. 113-147.

[63] Lucas R. (1978): Asset Prices in an Exchange Economy, *Econometrica*, 46, pp.1429-1445.

[64] MacMillan, L. W. (1986): Analytic Approximation for the American Put, *Advances in Futures and Options Research*, 1, pp. 119-139.

[65] Magill, M, and M Quinzii (1996): *Theory of Incomplete Markets*, Vol. 1, The MIT Press.

[66] McKean, H. P., Jr. (1965): Appendix: A Free Boundary Problem for the Heating Function Arising from a Problem in Mathematical Economics, *Industrial Management Review*, 6, pp. 32-39.

[67] Merton, R. (1976): Option Pricing when the Underlying Stock Returns are Discontinuous, *Journal of Financial Economics*, 5, pp. 125-144.

[68] Merton, R. (1990): *Continuous-Time Finance*, Blackwell.

[69] Mitchell, A., and D. F. Griffiths (2001): *Finite Difference and Related Methods for Differential Equations*, John Wiley & Sons.

[70] Myneni, R. (1992): The Pricing of American Option, *Annals of Applied Probability*, 2, pp. 1-23.

[71] Mulinacci, S. (1996): An Approximation of American Option Prices in a Jump-Diffusion Model, *Stochastic Processes and their Applications*, 62, pp. 1-17.

[72] Musiela, M, and M. Rutkowski (1997): *Martingale Methods in Financial Modelling*, Springer.

[73] Naik, V. and M. Lee (1990): General Equilibrium Pricing of Options on the Market Portfolio with Discontinuous Returns, *The Review of Financial Studies*, 3, pp. 493-521.

[74] Naik, V. (1993): Option Valuation and Hedging Strategies with Jumps in the Volatility of Asset Returns, *Journal of Finance*, 48, pp. 1969-1984.

[75] Omberg, E. (1988): Efficient Discrete Time Jump Process Models in Option Pricing, *Journal of Financial and Quantitative Analysis*, 23, pp. 161-174.

[76] Pan, J. (2002): The Jump-risk Premia Implicit in Options: Evidence from an Integrated Time-series Study, *Journal of Financial Economics*, 63, pp. 3-50.

[77] Peaceman, W., and H. Rachford (1955): The Numerical Solution of Parabolic and Elliptic Differential Equation, *Journal of Soc. Indust. Appl. Math.* III, pp. 28-42.

[78] Pham, H. (1997): Optimal Stopping, Free Boundary, and American Option in a Jump-Diffusion Model, *Applied Mathematics & Optimization*, 35, pp. 145-164.

[79] Press, W., S. Teukolsky, W. Vetterling, and B. Flannery (1997): *Numerical Recipes in C, The Art of Scientific Computing*, Second Edition, Cambridge University Press.

[80] Protter, P. (1990): *Stochastic Integration and Differential Equations*, Springer-Verlag.

[81] Rogers, L. and D. Talay (1997): *Numerical Methods in Finance*, Cambridge University Press.

[82] Saigal, R. (1970): A Note on a Special Linear Complementarity Problem, and Option Models, *Opsearch*, 7, pp. 175-183.

[83] Samuelson, P. A. (1965): Rational Theory of Warrant Pricing , *Industrial Management Review*, 6, pp. 13-31.

[84] Schäl, M. (1994): On quadratic Cost Criteria for Option Hedging, *Mathematics of Operations Research*, 19, pp. 121-131.

[85] Schweizer, M. (1991): Option Hedging for Semi-martingales, *Stochastic Processes and Their Applications*, 37, pp.339-363.

[86] Schweizer, M. (1995): Variance-optimal Hedging in Discrete-time, *Mathematics of Operations Research*, 20, pp. 1-32.

[87] Schweizer, M. (1996): Approximation Pricing and the Variance-optimal Martingale Measure, *Annals of Applied Probability*, 24, pp. 206-236.

[88] Scott, L. (1993): Pricing Stock Options in a Jump-Diffusion Model with Stochastic Volatility and Interest Rates: Applications of Fourier Inversion Methods, Department of Finance, University of Georgia.

[89] Tavella, D., and C. Randall (2000): *Pricing Financial Instruments: The Finite Difference Method*, John Wiley & Sons, Inc.

[90] Tilley, J. (1993): Valuing American Options in a Path Simulation Model, *Transactions of the Society of Actuaries*, 45, pp. 83-104.

[91] Villeneuve, S. (1999): Exercise Region of American Options on Several Assets, *Finance and Stochastics*, 3, pp. 295-322.

[92] Villeneuve, S., and A. Zanette (2002): Parabolic ADI Methods for Pricing American Options on Two Stocks, *Mathematics of Operations Research*, 27, pp. 121-149.

[93] Wiesenberg, H. (1998): Modeling Market Risk in a Jump-Diffusion Setting in a Generalized Hofmann-Platen-Schweizer-Model, Working paper, University of Bonn.

[94] Zariphopoulou, T. (1999): Optimal Investment and Consumption Models with Non-linear Stock Dynamics, *Mathematical Methods of Operations Research*, 50, pp. 271-296.

[95] Zariphopoulou, T. (2001): A Solution Approach to Valuation with Unhedgeable Risks, *Finance and Stochastics*, 5, pp. 61-82.

[96] Zhang, X. (1994): Méthodes numériques pour le calcul des options américaine dans des modéles de diffusion avec sauts, Thése de doctorat de l'Ecole Nationale des Ponts et Chaussées, Paris, 1994.

[97] Zhang, X. (1995): Formules quasi-explicites pour les options américaines dans un modéle de diffusion avec sauts, *Mathematics and Computers in Simulation*, 38, pp.151-161.

[98] Zhang, X. (1997): Numerical Analysis of American Option Pricing in a Jump-Diffusion Model, *Mathematics of Operations Research*, 22, pp. 668-690.

Vita

Byeongwook Choi was born in Incheon, Korea, on January 26, 1965, the son of Young Rock Choi and Soo Young Kim. After completing his High School studies in Seoul, Korea, in 1983, he entered the Seoul National University in Korea. He received degree of B.S. in Industrial Engineering from the University in 1987. After working about six years for Lucky-Gold Star Informations and Telecommunications, based in Seoul, he entered the MBA program in Rutgers, the State University of New Jersey and received an MBA degree in May of 1996. He moved to the University of Texas at Austin in the Fall of 1996, and since then has enrolled in the Ph.D. program in the Department of Management Science and Information Systems, the McCombs School of Business.

Permanent address: Yonsoo-gu, Yonsoo-dong 534,
Yoochun APT 102-901
Incheon, Korea

This dissertation was typeset with LaTeX[†] by the author.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.